TRANSIENT THERMAL STRESS ANALYSIS OF COMPOSITE STRUCTURES INCLUDING CONTINUOUSLY VARYING PROPERTIES

Richard Bailey Bubeck

NAVAL POSTGRADUATE SCHOOL
Monterey, California

THESIS

TRANSIENT THERMAL STRESS ANALYSIS
OF COMPOSITE STRUCTURES INCLUDING
CONTINUOUSLY VARYING PROPERTIES

by

Richard Bailey Bubeck

September 1975

Thesis Advisors: P. Pucci
G. Cantin

Approved for public release; distribution unlimited.

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) Transient Thermal Stress Analysis of Composite Structures Including Continuously Varying Properties		5. TYPE OF REPORT & PERIOD COVERED Engineer's Thesis; September 1975
		6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) Richard Bailey Bubeck		8. CONTRACT OR GRANT NUMBER(s)
9. PERFORMING ORGANIZATION NAME AND ADDRESS Naval Postgraduate School Monterey, California 93940		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS Naval Postgraduate School Monterey, California 93940		12. REPORT DATE September 1975
		13. NUMBER OF PAGES 344
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office)		15. SECURITY CLASS. (of this report) Unclassified
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Transient Thermal Stress Analysis

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

Calculations of transient thermal stress levels were made for composite coatings of hollow cylinders. Stress analysis was done during a thermal cycle of successive heating and cooling periods. Various coating configurations were investigated. The results showed the effect of changes in coating arrangements and the failure modes due to the excessive

UNCLASSIFIED

REPORT DOCUMENTATION PAGE

4. TITLE (and Subtitle)

Transient Thermal Stress Analysis of Composite Propellant Grains [illegible]

5. TYPE OF REPORT & PERIOD COVERED

[illegible] Thesis; [illegible]

7. AUTHOR(s)

Richard [illegible]

9. PERFORMING ORGANIZATION NAME AND ADDRESS

Naval Postgraduate School
Monterey, California 93940

11. CONTROLLING OFFICE NAME AND ADDRESS

Naval Postgraduate School
Monterey, California 93940

12. REPORT DATE

[illegible] 1979

13. NUMBER OF PAGES

[illegible]

15. SECURITY CLASS. (of this report)

Unclassified

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Transient Thermal Stress Analysis

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

[illegible]

DD FORM 1473 EDITION OF 1 NOV 65 IS OBSOLETE

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

(20. ABSTRACT Continued)

stress levels.

Solutions were obtained by approximate techniques, and a computer program for transient three dimensional thermal stress analysis was developed. The finite element technique was employed for the determination of the temperature distribution and the elastic stress analysis. Fortran IV G Level was used as the programming language.

Transient Thermal Stress Analysis
of Composite Structures Including
Continuously Varying Properties

by

Richard Bailey Bubeck
Lieutenant, United States Navy
B.A., University of Virginia, 1967

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN MECHANICAL ENGINEERING

and

MECHANICAL ENGINEER

from the

NAVAL POSTGRADUATE SCHOOL
September 1975

ABSTRACT

Calculations of transient thermal stress levels were made for composite coatings of hollow cylinders. Stress analysis was done during a thermal cycle of successive heating and cooling periods. Various coating configurations were investigated. The results showed the effect of changes in coating arrangements and the failure modes due to the excessive stress levels.

Solutions were obtained by approximate techniques, and a computer program for transient three dimensional thermal stress analysis was developed. The finite element technique was employed for the determination of the temperature distribution and the elastic stress analysis. Fortran IV G Level was used as the programming language.

TABLE OF CONTENTS

Part One

Thermal Stress Analysis of Cylinders With Composite Coatings

Part Two

Transient Thermal Stress Analysis

ACKNOWLEDGMENTS

The author wishes to express his appreciation to Professor Paul F. Pucci and Professor Gilles Cantin for their advice and guidance during the development of this project. Without their valuable assistance, this thesis could not have been completed. In addition, the author wishes to thank Professor Thomas E. Cooper for his comments and advice.

The author wishes to thank Mr. John W. Fairbanks of NAVSEC, Hyattsville, Md., and Mr. Ray Busch of Battelle Pacific Northwest Laboratories, Richland, Washington for their recommendations and suggestions regarding the project undertaken.

Finally, the author wishes to thank his wife, Donnalee, for her understanding and encouragement throughout the course of this study.

PART I. THERMAL STRESS ANALYSIS OF CYLINDERS WITH COMPOSITE COATINGS

The efficiency of a gas turbine engine can be increased by raising the turbine inlet temperature. However, protection against high temperature oxidation or sulfidation of the turbine blades must be provided. One method, which has been suggested, [1]*, is the application of ceramic coatings which resist high temperature corrosion. Since the material properties vary through such a composite blade, thermal stresses become an important consideration particularly during transients with non-uniform temperature distributions.

The application of these ceramic coatings has been accomplished via a process of high-rate sputtering [1]. This process requires high temperatures to insure proper adhesion of the coatings. Since the thermal life cycle of the coatings differs from the thermal conditions when the coatings are applied, the deposition temperature must be considered in the design criteria.

In order to determine the thermal stress levels within possible coating arrangements, hollow cylinders were analyzed with various material compositions. A thermal cycle consisting of a rapid heating phase followed by a cooling phase was used to generate temperature fluctuations and the thermal

---

* Numbers in brackets refer to similarly numbered references in the bibliography.

stress distributions. The finite element technique was used to perform the elastic stress analysis and to determine the temperature distributions.

I. SIMULATION OF THE EXPERIMENTAL MODEL

An experimental program [1] to develop the coating technique and to observe the coating during thermal transients was begun using thin walled cylinders, a simple geometry instead of the more complex turbine blade. Such cylinders were prepared by the high-rate sputter deposition of nickel and stabilized zirconia on a nickel superalloy substratum. In addition to the pure coatings, a graded composition layer, composed of nickel and zirconia, was used between the substratum and the outer layer. The cylinders were exposed to a thermal cycle consisting of a rapid heating phase followed by a cooling phase. The process was accomplished by the cross-flow of high temperature air past the hollow cylinder, followed by the use of low temperature air. The analysis reported herein attempts to analytically simulate this experiment. In doing so, several simplifying assumptions were made.

The gas flow around the cylinder was considered symmetric, therefore, only one half of the hollow cylinder was modelled. A plane, along the cylinder axis, passing through the forward stagnation point and the after stagnation point was considered an adiabatic surface. No axial variation of the heat transfer conditions was allowed, however, the local heat transfer conditions were varied in the circumferential directions.

Although during an actual experiment the cylinder was supported by some apparatus, the modelled cylinder was unsupported. Therefore, all transfer of heat occurred only in the hollow cylinder, and it was free to expand or contract in all directions.

The simulation analysis was used to determine the temperature distribution and the stress distribution present in the cylinder at various times throughout the thermal cycle. Excessive stress levels were then used to predict when and what type of failure would occur.

The classical thermal stress equations are coupled to the heat equation, however, during this simulation analysis, the heat equation was solved separately. The temperature distribution was then used to solve the thermal stress problem.

II. CYLINDER CONFIGURATIONS

Four configurations were analyzed to determine the stress levels at various times in the thermal cycle. All geometries consisted of a hollow cylinder three inches long, an inside diameter of 0.5 inches, and an outside diameter dependent on the particular coating used. The thickness of the nickel superalloy substratum was 0.05 inches, and the thickness of the coatings varied from 0.005 to 0.009 inches. Drawings of all configurations are shown in Figure 1.

A. CONFIGURATION NUMBER ONE

The first coating investigated was a 0.005 inch layer of zirconia on the nickel superalloy. Material properties changed from those of the nickel superalloy to those of the zirconia coating at 0.3 inches radius.

The finite element mesh used in this arrangement consisted of 21 isoparametric quadratic elements totaling 200 nodes. The element discretization was specified as one element in the axial direction, three elements in the circumferential direction, and seven elements in the radial direction. The seven radial elements were proportioned into four elements through the substratum and three elements through the layer.

B. CONFIGURATION NUMBER TWO

The second arrangement analyzed consisted of 0.002 inches of pure nickel on the substratum followed by 0.005 inches of

stabilized zirconia. The material properties abruptly changed at two radial positions.

The finite element mesh for this configuration consisted of 21 isoparametric quadratic elements using 200 nodes. The subdivision in the axial and circumferential directions remained as in configuration number one. The seven radial elements were arranged with three elements in the substratum and two elements in each layer.

C. CONFIGURATION NUMBER THREE

The third arrangement analyzed consisted of 0.004 inches of graded layer on the substratum. This graded layer allowed a gradual change of the material properties from those of nickel to those of zirconia. This arrangement was used to match the material properties of the substratum to those of the coatings. A 0.003 inch layer of stabilized zirconia was placed on top of the graded layer. Properties of the graded composition varied in a linear fashion as described in Section III.

This configuration was discretized into 24 finite elements totaling 226 nodes. The axial and circumferential directions were divided into one element and three elements respectively. The eight elements in the radial direction were arranged with four elements in the substratum, two elements in the graded layer, and two elements in the outer layer of zirconia.

D. CONFIGURATION NUMBER FOUR

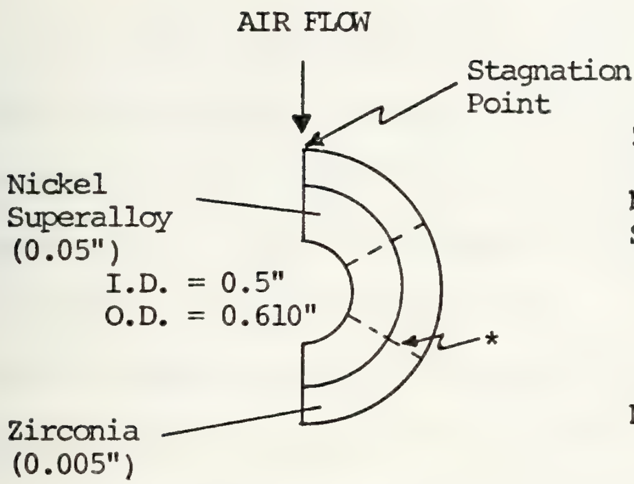
The fourth configuration was composed of the substratum followed by a 0.002 inch layer of pure nickel. A 0.004 inch coating of graded composition was placed on the pure nickel. A 0.003 inch coating of pure zirconia was used as the outer layer.

The finite element mesh for this configuration consisted of 27 isoparametric quadratic elements using 252 nodes. The axial and circumferential directions were subdivided as in the other configurations. Nine elements were used in the radial direction, of which three were placed in the substratum, followed by two in each of the three layers.

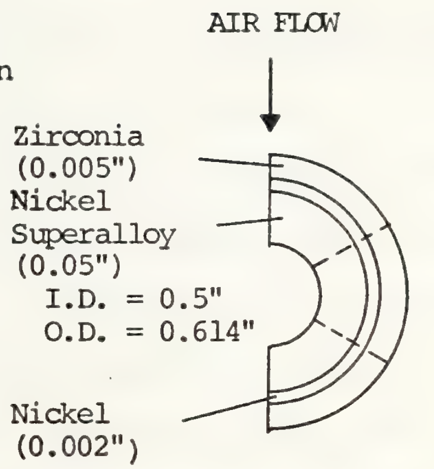
E. COMMENTS ON THE FINITE ELEMENT MESH

Emphasis has been placed on the radial and tangential variations, therefore, the mesh was made finer for those directions. Since a limitation on the computer core requirement and computing time dictates the number of elements and nodes, the axial direction was modelled using only one element. The problem solution is a three dimensional method, therefore, ideally it would be more appropriate to have additional elements. Since the main interest was focused on the radial variation in material composition and the circumferential variation of heat transfer coefficients, the mesh used was considered appropriate. There was no axial variation in material properties or boundary conditions, therefore, no axial variation in the temperature distribution resulted.

A slight variation in the stress distribution resulted due to the three dimensional nature of the problem.

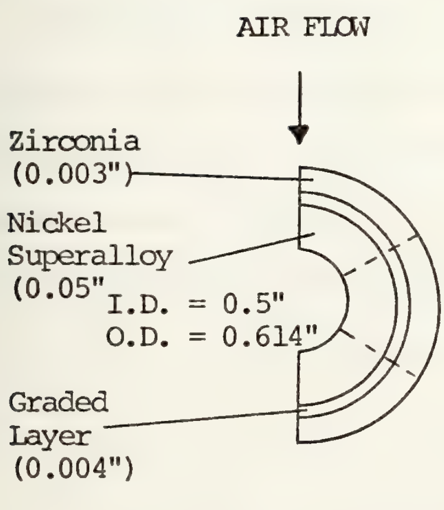


Configuration No. 1

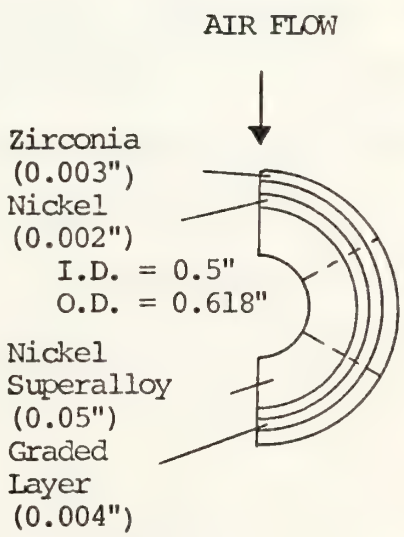


Configuration No. 2

(*) Radial dashed lines indicate element boundaries in the circumferential direction.



Configuration No. 3



Configuration No. 4

FIGURE 1. CYLINDER CONFIGURATIONS

III. MATERIAL PROPERTIES

The materials used in the analysis were a nickel-superalloy (IN-738), pure nickel, and stabilized zirconia ($Zr\ O_2$ plus $Ca\ O$). The substratum of the hollow cylinder was composed of the nickel superalloy, and the coatings were combinations of nickel and zirconia. Material property values were assumed constant throughout the temperature range investigated. The values used were the average of those available in the literature [2,3]. The numerical values for elastic modulus, Poisson's ratio, coefficient of linear expansion, thermal conductivity, and volumetric heat capacity (density times specific heat) of the various materials are given in Table 1.

Matching of material properties was attempted by using a graded composition between layers in some of the coating configurations. Approximation to the properties of these compositions was made by using a function based on the radial position. If two materials A and B have properties P_A and P_B, the property of the composition P_C was determined by

$$P_C = P_A\left(\frac{r_B - r}{r_B - r_A}\right) + P_B\left(\frac{r - r_A}{r_B - r_A}\right) \qquad (1)$$

where r_A is the radius of material A and r_B, that of material B.

TABLE 1

MATERIAL PROPERTIES

Nickel Superalloy - IN-738

Modulus of Elasticity	- 31.75×10^{6} PSI
Poisson's Ratio	- 0.3
Coefficient of Linear Expansion	- 8.6×10^{-6} in/in-deg.F
Thermal Conductivity	- 1.34×10^{-2} BTU/min-in-deg.F
Volumetric Heat Capacity	- 0.029 BTU/cu.in.-deg.F

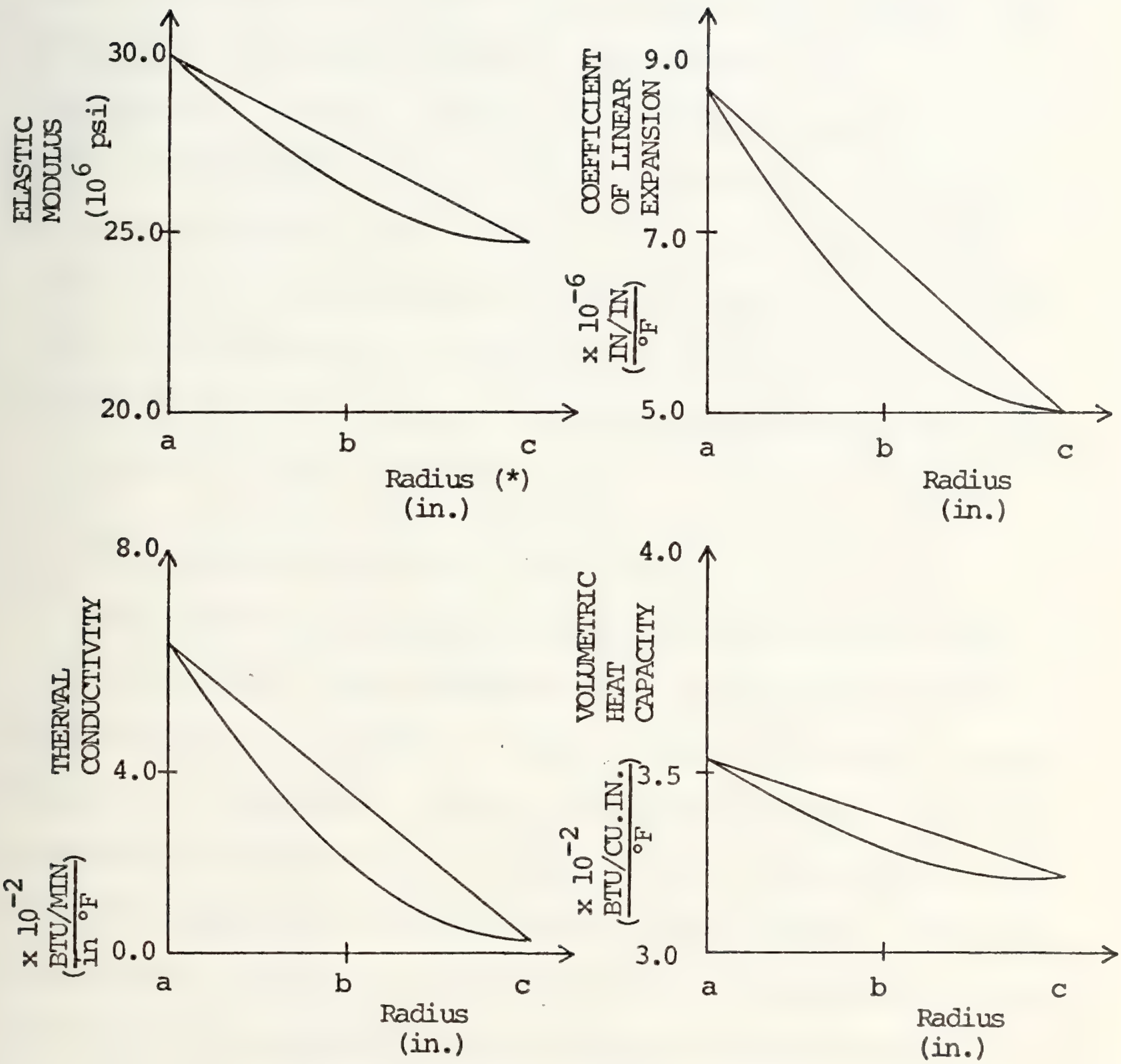
Pure Nickel

Modulus of Elasticity	- 30.00×10^{6} PSI
Poisson's Ratio	- 0.3
Coefficient of Linear Expansion	- 8.6×10^{-6} in/in-deg.F
Thermal Conductivity	- 6.89×10^{-2} BTU/min-in-deg.F
Volumetric Heat Capacity	- 0.035 BTU/cu.in.-deg.F

Stabilized Zirconia

Modulus of Elasticity	- 24.8×10^{6} PSI
Poisson's Ratio	- 0.3
Coefficient of Linear Expansion	- 5.0×10^{-6} in/in-deg.F
Thermal Conductivity	- 3.36×10^{-3} BTU/min-in-deg.F
Volumetric Heat Capacity	- 0.032 BTU/cu.in.-deg.F

The materials used in the graded layer were pure nickel and stabilized zirconia. The values for the various properties of the graded compositions are shown in the material property graphs, Figure 2. The property values used during the quadratic variations were determined by specifying property values for three points in the layer and then forming the quadratic function. Two of the points were specified to have properties of the inside and outside of the graded layer, i.e., the properties of nickel and zirconia. The midpoint was specified as the arithmetic mean of the properties at the midpoint of the linear variation and the properties of zirconia. The resultant quadratic variation is shown on the same plots as the linear variation.



(*) Configuration Three: a=0.3, b=0.302, c=0.304
Configuration Four : a=0.302, b=0.304, c=0.306

FIGURE 2. MATERIAL PROPERTY GRAPHS

IV. THERMAL CYCLE CONDITIONS

The conditions which specify the thermal stress problem are classified as boundary conditions and initial conditions. With respect to the temperature distribution problem, the boundary conditions consist of convection heat transfer coefficients and the external fluid temperatures, and the initial condition is the initial temperature distribution. The stress problem has boundary conditions which specify zero or given displacement at various positions. In addition, the zero-stress temperature distribution must be supplied since all temperature distributions are compared to the zero stress condition.

A. HEAT TRANSFER BOUNDARY CONDITIONS

The thermal cycle consisted of a rapid heating phase and a cooling phase. The cylinder wall was heated from room temperature (77 deg. F) to approximately 2000 deg. F in a period of 45 seconds. During the cooling phase, the heat transfer conditions were changed to cool the cylinder back to room temperature in approximately two minutes. During the heating phase, a convection boundary condition was maintained on the external surface and an adiabatic condition on the internal surface. The cooling phase was characterized by convection boundary conditions on both internal and external surfaces. A circumferential variation of the

external heat transfer coefficients was used during both phases. The internal heat transfer coefficient was uniform during the cooling phase. No axial variation in the heat transfer coefficients was used.

In order to determine an average heat transfer coefficient for both the heating and cooling phases of the transient cycle, the cylinder was treated thermally as a lumped capacitor. Using this average coefficient, an average Nusselt number was calculated using air as the external heating fluid. The temperature of the external fluid was assumed to be 2500 °F. A Reynolds number for flow past a cylinder was then computed using the Hilpert correlation. This Reynolds number was used to determine the local Nusselt number and therefore heat transfer coefficient for air flow past a horizontal cylinder. These values of Nusselt number were obtained from Kreith [5] and Krall and Eckert [6]. For the cooling phase, the average heat transfer coefficient was used for the internal surface. The values used during the computer analysis are shown in Table 2. These values were used for all configurations.

The resultant temperatures of the cylinder wall were considered adequate for the analysis. The temperature variation at 45 seconds was between 1850 deg.F and 2050 deg F. Room temperature (77 deg. F) was reached after 2.25 minutes into the cooling phase.

TABLE 2

Heating Phase

Fluid Temperature: 2500 deg. F
Average Heat Transfer Coefficient: 22.6 BTU/hr-ft^2-deg. F
Average Nusselt Number: 22.5
Reynolds Number: 2262
Equivalent Fluid Velocity: 124.6 ft/sec

Local Heat Transfer Coefficients

Angle Position from Stagnation (degrees)	Coefficient (BTU/hr-ft^2-deg. F)
0.0	40.0
30.0	41.0
60.0	38.0
90.0	26.0
120.0	11.5
150.0	16.0
180.0	12.0

Cooling Phase

Fluid Temperature: 60 deg. F
Average Heat Transfer Coefficient: 13.5 BTU/hr-ft^2-deg. F
Average Nusselt Number: 47.1
Reynolds Number: 7300
Equivalent Fluid Velocity: 22.9 ft/sec

Local Heat Transfer Coefficients

Angle Position from Stagnation (degrees)	Coefficient (BTU/hr-ft^2-deg. F)
0.0	22.0
30.0	20.5
60.0	15.0
90.0	9.0
120.0	6.0
150.0	12.0
180.0	12.0

B. STRESS ANALYSIS BOUNDARY CONDITIONS

The hollow cylinders analyzed were free to expand or contract in all directions.

C. INITIAL CONDITIONS

The cylinder was specified to be at an initial uniform temperature of 77 deg. F. The zero-stress temperature distribution was assumed to be the temperature when the coatings were applied. This was specified to be 932.0 deg. F.

V. RESULTS OF THE ANALYSIS

The results of the analysis showed that the stresses within the surface coatings varied from compression at room temperature to tension at the maximum temperature. Due to the variation of the heat transfer boundary conditions, a uniform temperature equal to the zero-stress condition was not present during the thermal cycle. The complex nature of the circumferential variation in heat transfer coefficients causes temperature and stress distributions vastly different from those common to radial heat flow.

All configurations follow the same general stress history. During the period when the temperatures are below the zero-stress condition, the coatings remain in compression. At the same time, the sign of the substratum stresses varies according to the position. At approximately 12 seconds, the temperature distributions reached the zero-stress condition and the stresses in the coatings became tensile. These tensile stresses increased as the temperature increased reaching a maximum at 0.75 minutes. All configurations followed this general scheme. The maximum tensile stress is dependent on the zero-stress condition, i.e., the higher the zero-stress temperature, the lower the maximum tensile stress. Configuration Four had somewhat lower stress levels than the other configurations; however, the difference is not considered significant.

The use of a graded layer was investigated further by specifying a quadratic composition instead of a linear composition. An adjustment of the thermal conductivity and the coefficient of linear expansion to follow a quadratic function produced only minor changes in the stress levels. When all properties were adjusted to the quadratic composition variation, no appreciable change occurred in the stress levels.

Although the maximum stress levels are excessive, the maximum strains (9.6×10^{-3}) are considered small enough to justify the use of a linear elastic analysis. In addition, the brittle nature of zirconia justifies the use of a brittle fracture failure criteria. Therefore, the elastic analysis provides a reliable method for determining whether the structure will fail or not. Using the maximum principal stress theory of failure, some prediction of the time and temperature when failure occurs can be made. If the ultimate strength of the ceramic material is set at 18.0 KSI and that of the nickel superalloy at 90.0 KSI, observation of the average tangential and axial stress levels for all configurations indicates that failure would occur at about 0.2 minutes and an average temperature of 1000.0 °F. At most, this is only a rough calculation. Observations of an actual specimen could confirm this estimate by raising and lowering the specimen to successively higher temperatures, followed by appropriate microscopic examination.

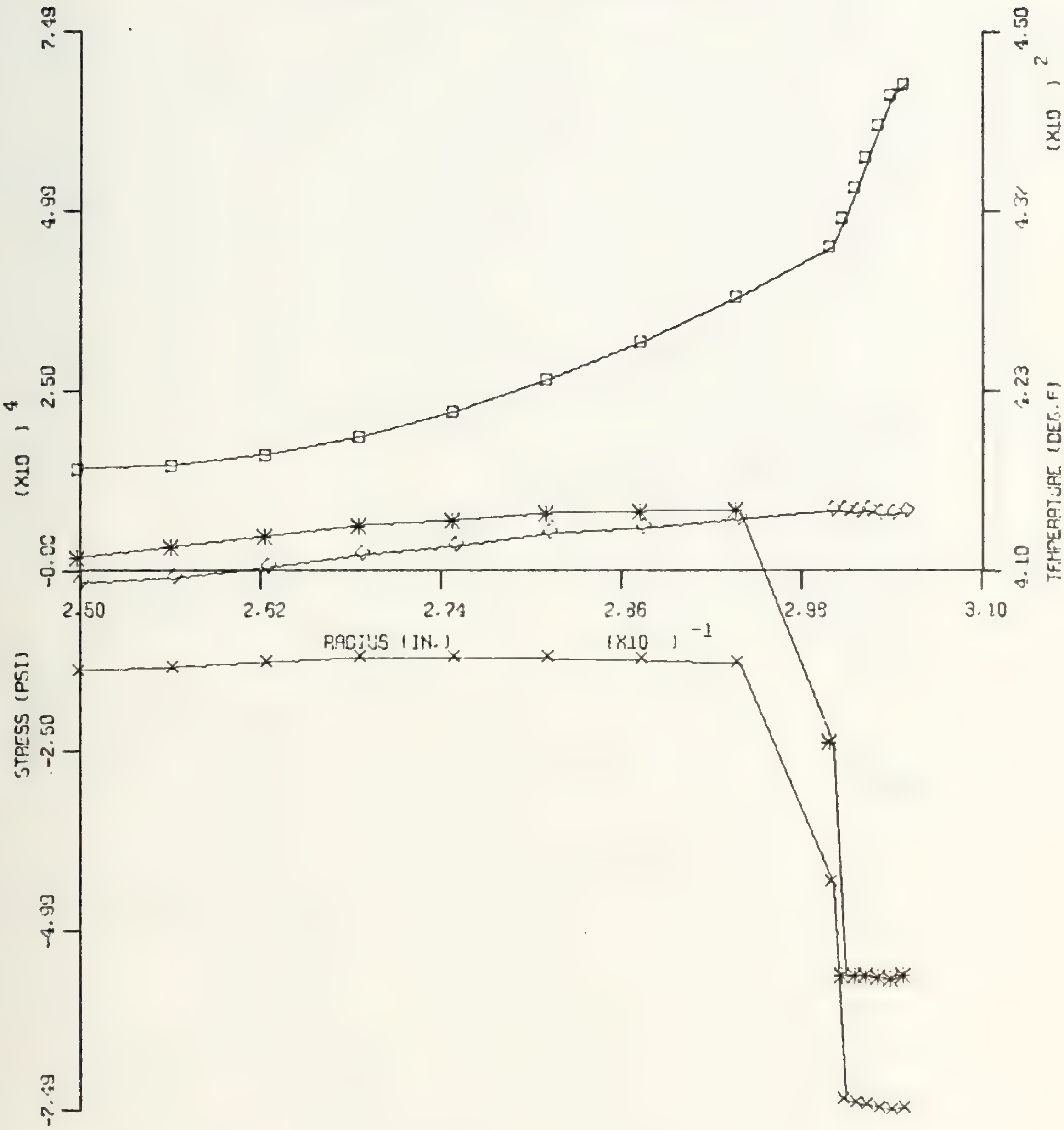
The graphical representations of the thermal stress and temperature distributions were made using the numerical values obtained for the nodes at the cylinder ends. No attempt was made to smooth out the curves and no interpolations were performed. Each plot is identified by a time and a configuration designation. For example, the first plot is identified by TIME = 0.05 CONF.1-0. This plot represents the conditions at 0.05 minutes after start of the thermal cycle for configuration number one at position zero degrees relative to the stagnation point. Each axis has been labelled and the units identified. Since the range of values for stress and temperature varies from one graph to another, the scale factors used in plotting are provided on each plot. In order to provide an overall view of the stress levels throughout the thermal cycle, a plot of extreme principal stresses versus time has been prepared for each configuration, Figures 7, 14, 21, 26.

A. CONFIGURATION NUMBER ONE

This configuration contained a single layer of stabilized zirconia on the nickel superalloy substratum. Comparison of the maximum principal stresses in all four configurations shows that configuration number one has the highest stress level. This maximum stress level (140.0 KSI) occurred at the maximum temperature, i.e., at the end of the heating phase, and was located at a radius of 0.3017 inches, the

zero angle position, and at the midpoint of the cylinder length. The largest tensile stresses present in configuration number one were all tangential and axial components of the stress state. The following figures show the radial variation of temperature and stresses at a given angular position for selected times during the heat transfer cycle: Figure 3 (a through j) at zero degrees, Figure 4 (a through j) at 60 degrees, Figure 5 (a through j) at 120 degrees, Figure 6 (a through j) at 180 degrees. (Similar sets of figures are provided for the remaining three configurations.)

STRESS AND TEMPERATURE
VERSUS RADIAL POSITION
TIME = 0.050 CONF.1-0



SIGMA R - DIAMOND SIGMA T - ASTERISK
SIGMA Z - CROSS TEMPERATURE - SQUARE

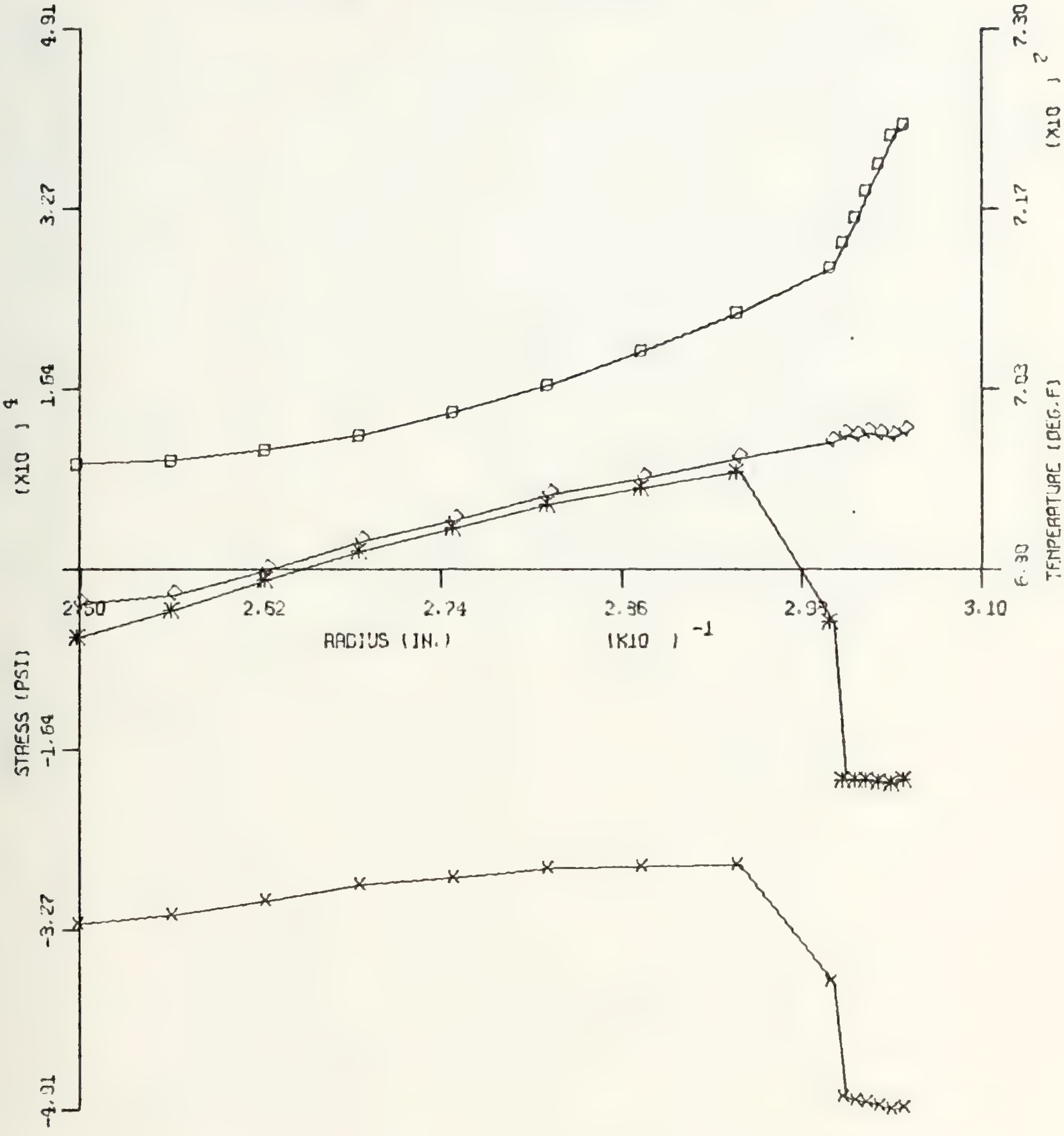
STRESS SCALE = $2.496X\ 10^{4}$ PSI/INCH

TEMPERATURE SCALE = 13.3 DEGREES/INCH

RADIAL POSITION SCALE = 0.01 INCHES/INCH

FIGURE 3(a)

STRESS AND TEMPERATURE
VERSUS RADIAL POSITION
TIME = 0.100 CONF.1-0



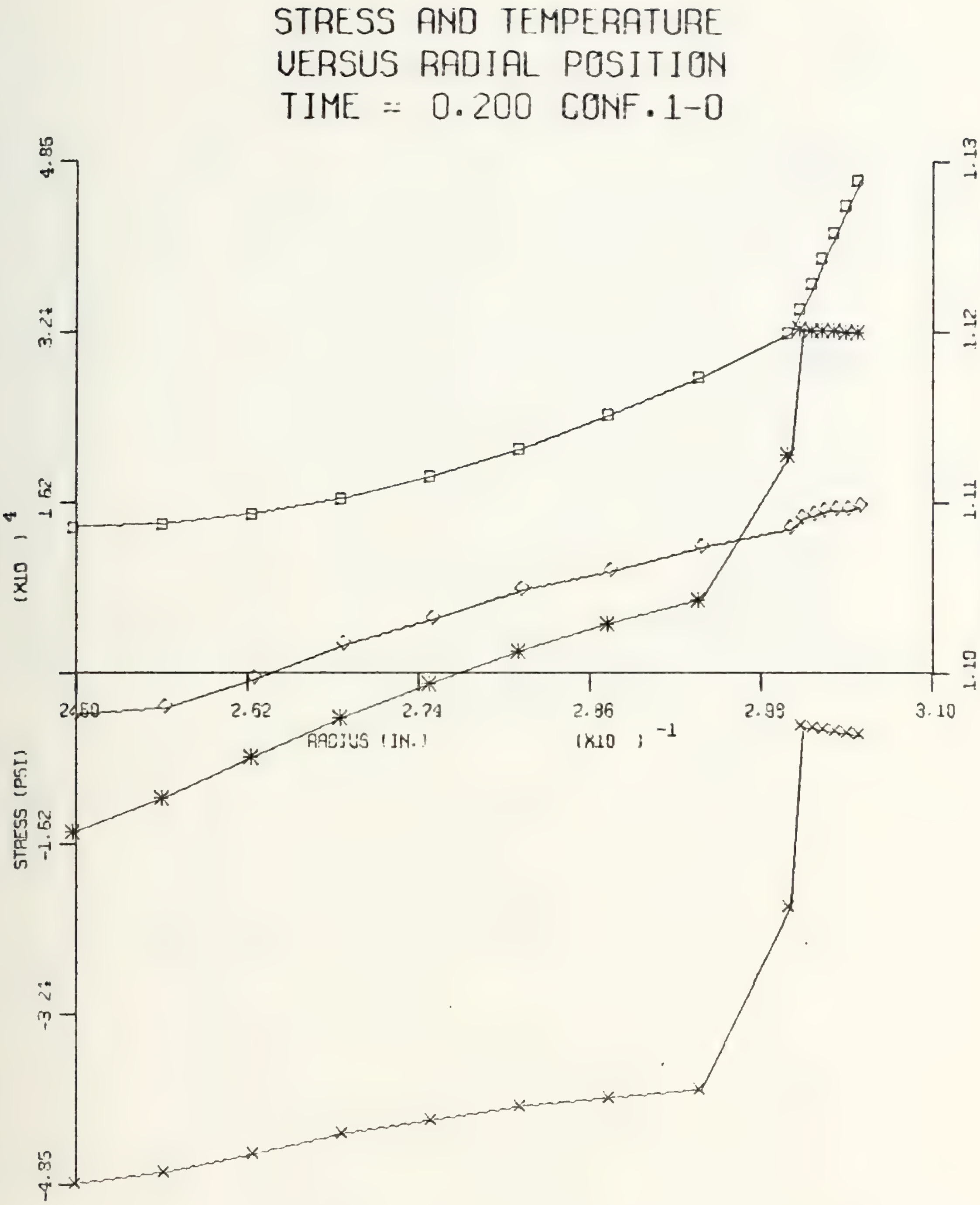
SIGMA R - DIAMOND SIGMA T - ASTERISK
SIGMA Z - CROSS TEMPERATURE - SQUARE

STRESS SCALE = 1.635X 10^4 PSI/INCH

TEMPERATURE SCALE = 13.3 DEGREES/INCH

RADIAL POSITION SCALE = 0.01 INCHES/INCH

FIGURE 3(b)



SIGMA R - DIAMOND SIGMA T - ASTERISK
SIGMA Z - CROSS TEMPERATURE - SQUARE

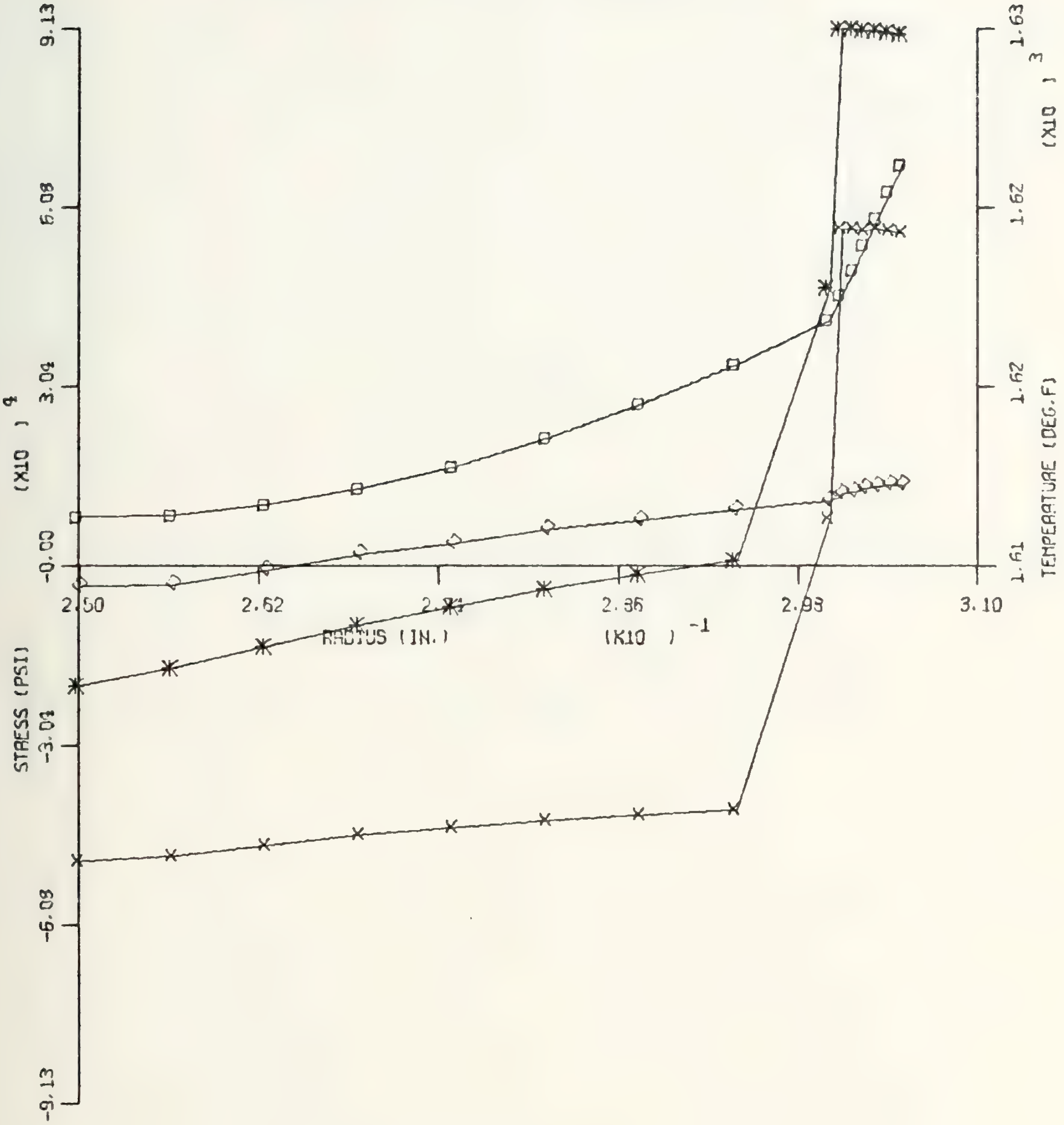
STRESS SCALE = 1.618X 10^{4} PSI/INCH

TEMPERATURE SCALE = 10.0 DEGREES/INCH

RADIAL POSITION SCALE = 0.01 INCHES/INCH

FIGURE 3(c)

STRESS AND TEMPERATURE
VERSUS RADIAL POSITION
TIME = 0.400 CONF.1-0



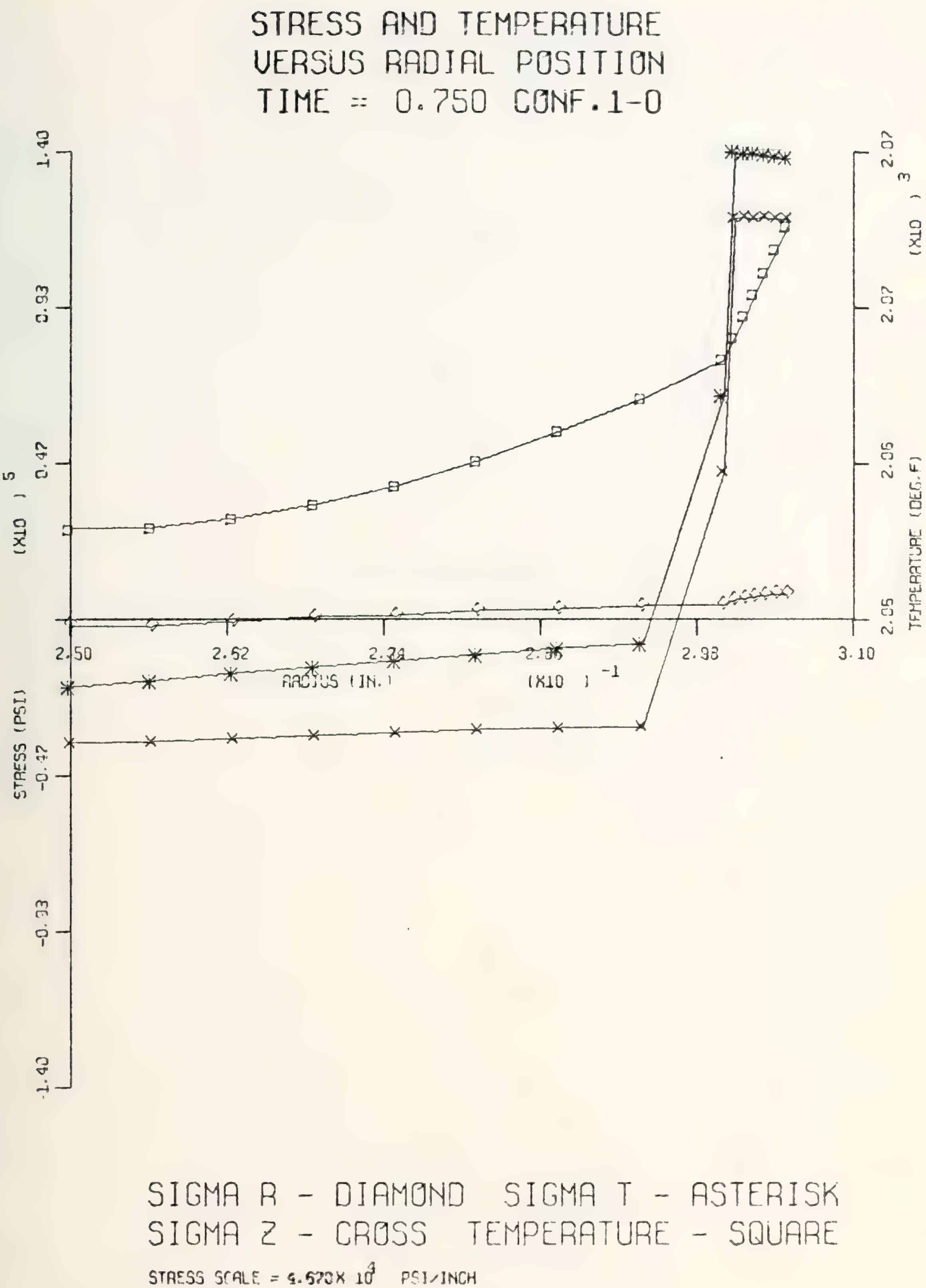
SIGMA R - DIAMOND SIGMA T - ASTERISK
SIGMA Z - CROSS TEMPERATURE - SQUARE

STRESS SCALE = 3.042X 10[4] PSI/INCH

TEMPERATURE SCALE = 5.7 DEGREES/INCH

RADIAL POSITION SCALE = 0.01 INCHES/INCH

FIGURE 3(d)



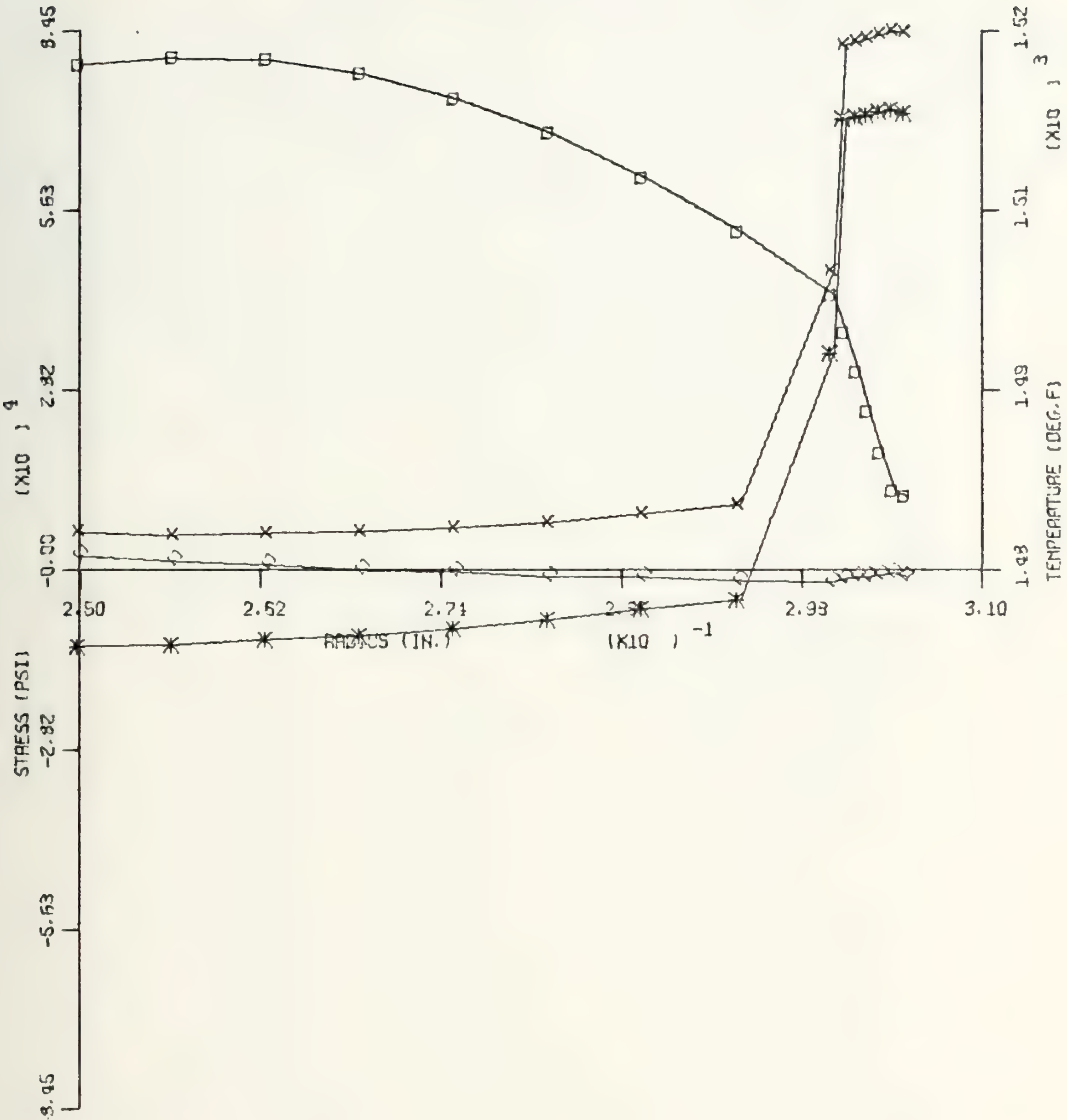
STRESS SCALE = 4.670X 10^4 PSI/INCH

TEMPERATURE SCALE = 3.3 DEGREES/INCH

RADIAL POSITION SCALE = 0.01 INCHES/INCH

FIGURE 3(e)

STRESS AND TEMPERATURE
VERSUS RADIAL POSITION
TIME = 0.800 CONF.1-0



SIGMA R - DIAMOND SIGMA T - ASTERISK
SIGMA Z - CROSS TEMPERATURE - SQUARE

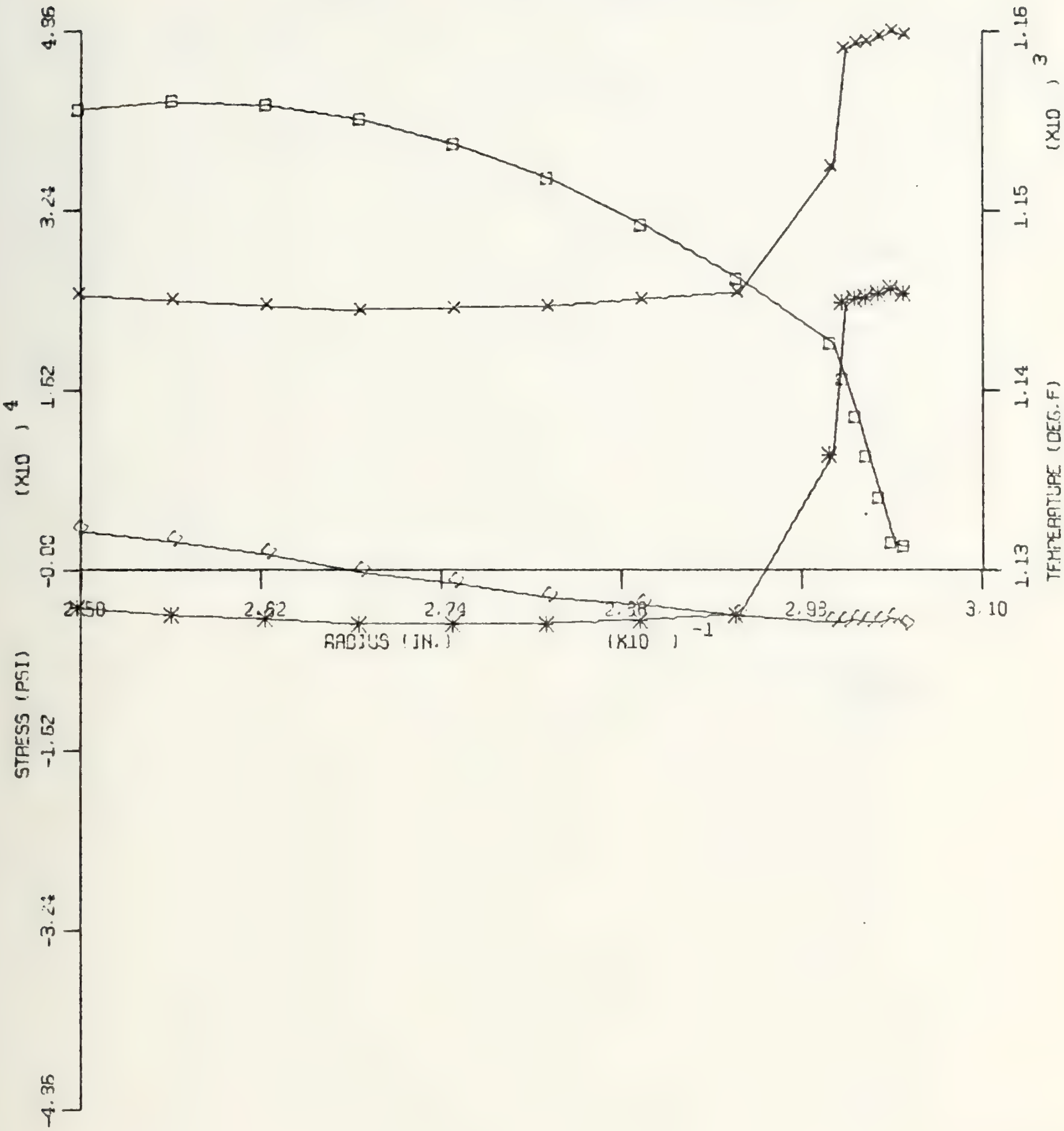
STRESS SCALE = 2.817X 10^{4} PSI/INCH

TEMPERATURE SCALE = 13.3 DEGREES/INCH

RADIAL POSITION SCALE = 0.01 INCHES/INCH

FIGURE 3(f)

STRESS AND TEMPERATURE
VERSUS RADIAL POSITION
TIME = 0.850 CONF.1-0



SIGMA R - DIAMOND SIGMA T - ASTERISK
SIGMA Z - CROSS TEMPERATURE - SQUARE

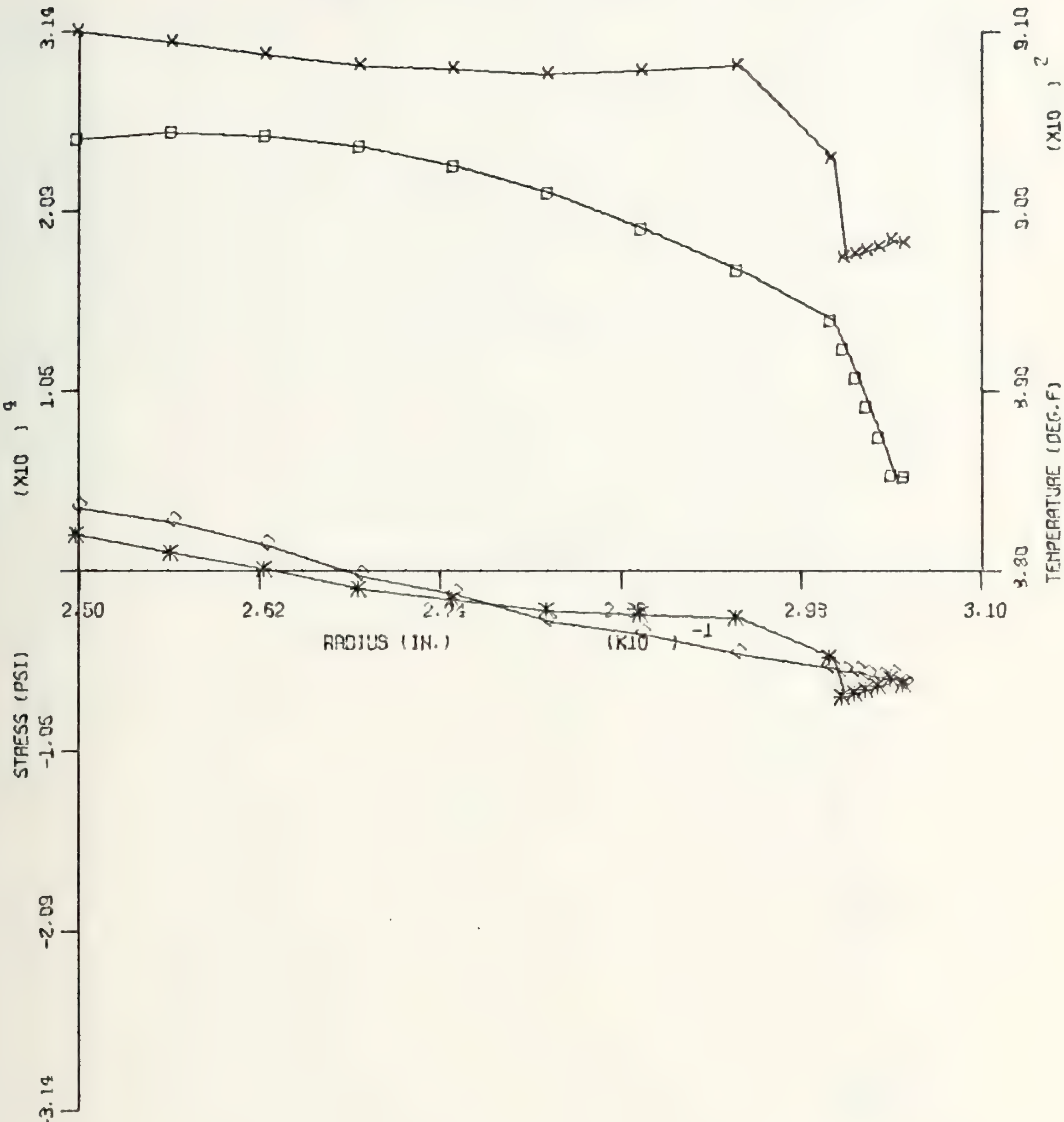
STRESS SCALE = 1.619X 10^4 PSI/INCH

TEMPERATURE SCALE = 10.0 DEGREES/INCH

RADIAL POSITION SCALE = 0.01 INCHES/INCH

FIGURE 3(g)

STRESS AND TEMPERATURE
VERSUS RADIAL POSITION
TIME = 0.900 CONF.1-0



SIGMA R - DIAMOND SIGMA T - ASTERISK
SIGMA Z - CROSS TEMPERATURE - SQUARE

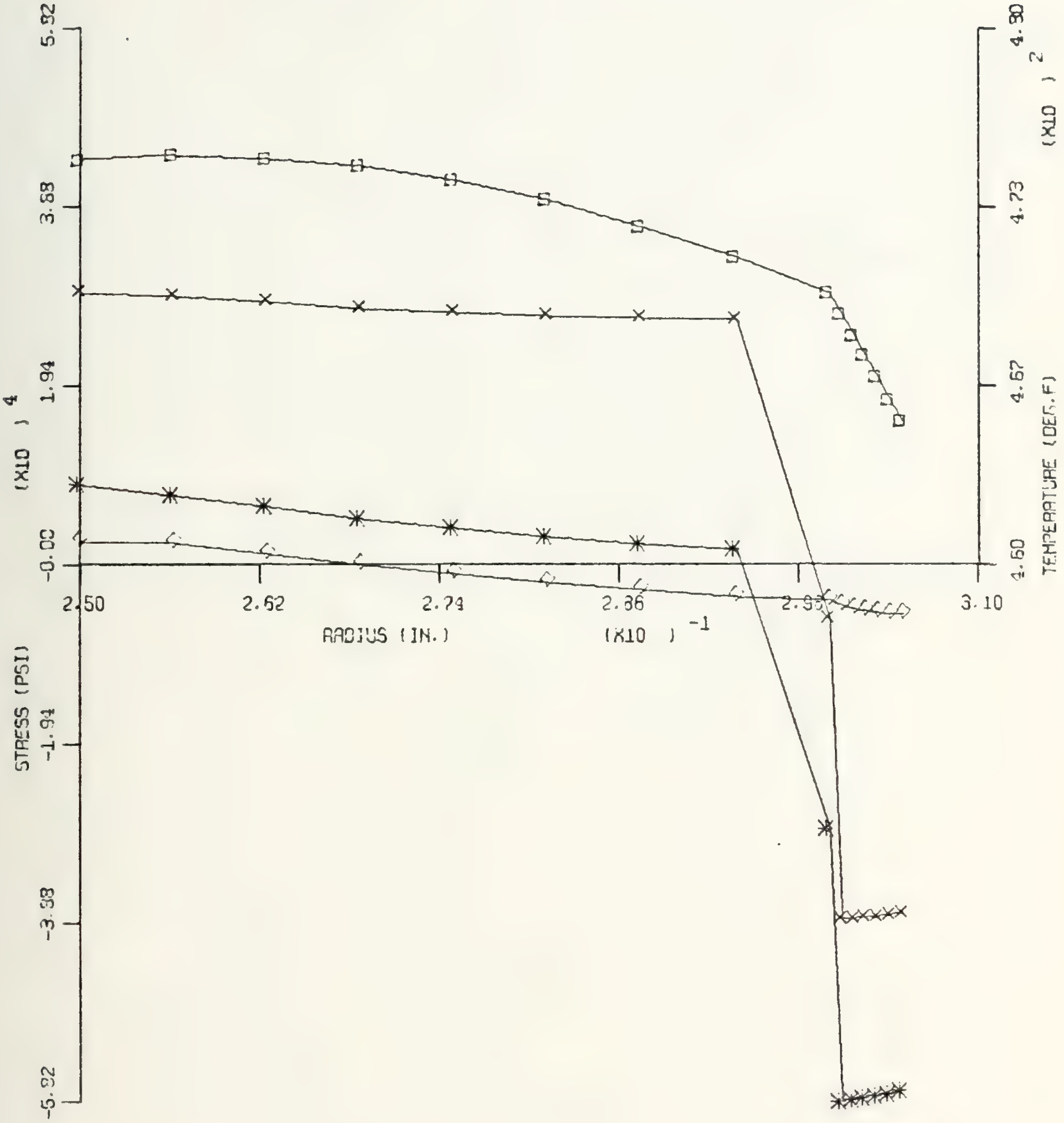
STRESS SCALE = 1.046 X 10^4 PSI/INCH

TEMPERATURE SCALE = 10.0 DEGREES/INCH

RADIAL POSITION SCALE = 0.01 INCHES/INCH

FIGURE 3(h)

STRESS AND TEMPERATURE
VERSUS RADIAL POSITION
TIME = 1.050 CONF.1-0



SIGMA R - DIAMOND SIGMA T - ASTERISK
SIGMA Z - CROSS TEMPERATURE - SQUARE

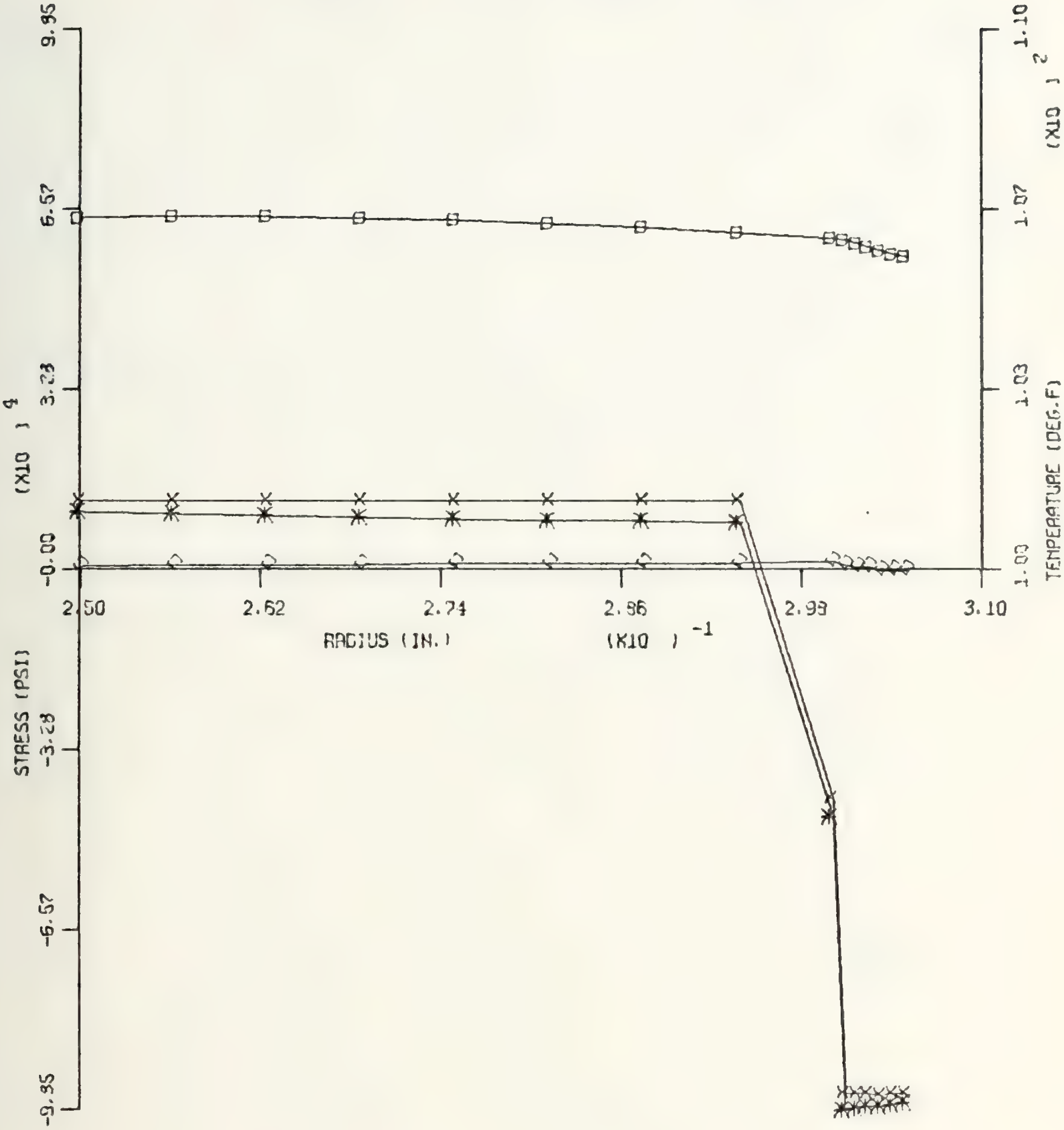
STRESS SCALE = 1.911X 10[4] PSI/INCH

TEMPERATURE SCALE = 6.7 DEGREES/INCH

RADIAL POSITION SCALE = 0.01 INCHES/INCH

FIGURE 3(i)

STRESS AND TEMPERATURE
VERSUS RADIAL POSITION
TIME = 1.650 CONF.1-0



SIGMA R - DIAMOND SIGMA T - ASTERISK
SIGMA Z - CROSS TEMPERATURE - SQUARE

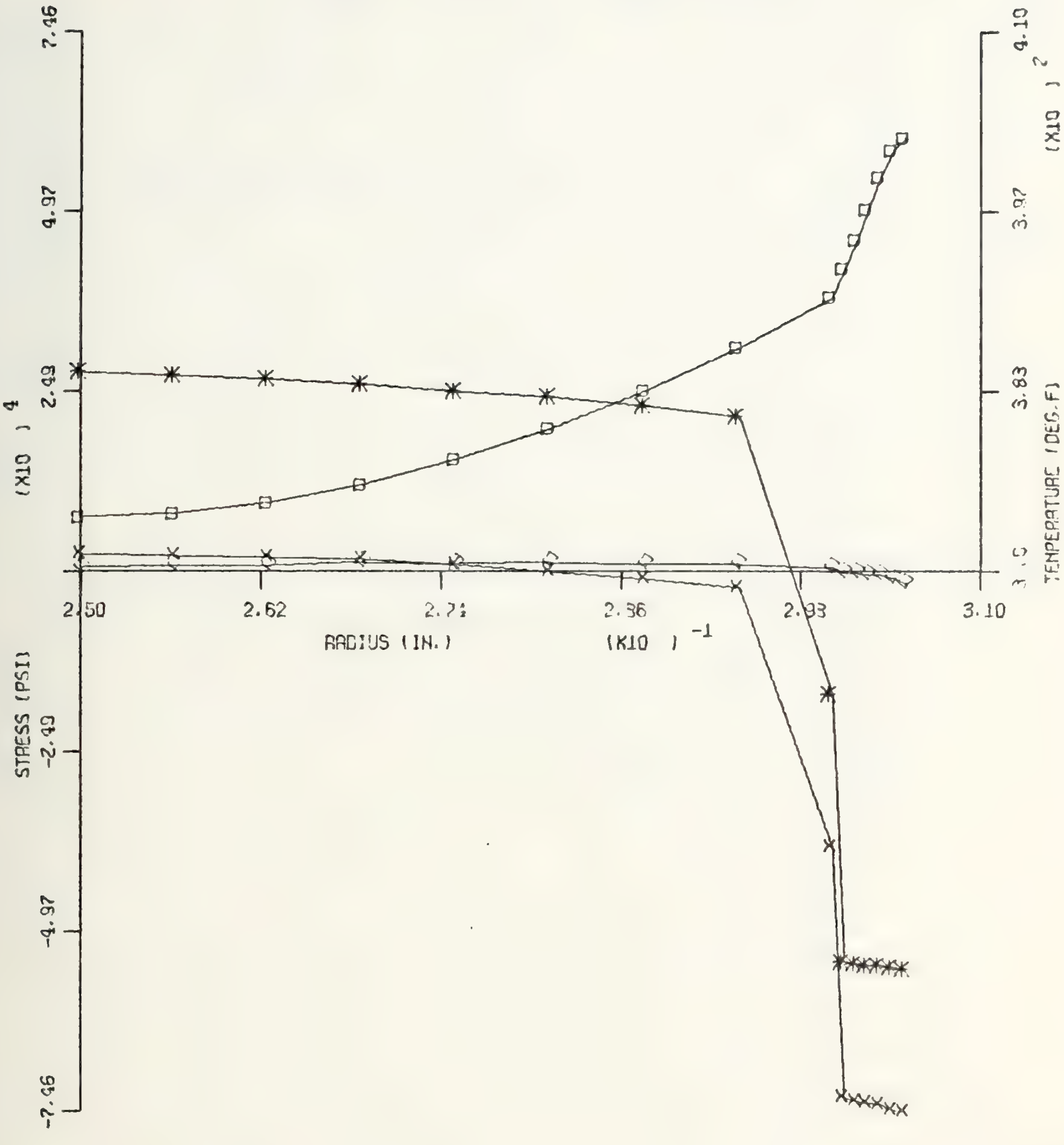
STRESS SCALE = 3.283X 10⁴ PSI/INCH

TEMPERATURE SCALE = 3.3 DEGREES/INCH

RADIAL POSITION SCALE = 0.01 INCHES/INCH

FIGURE 3(j)

STRESS AND TEMPERATURE
VERSUS RADIAL POSITION
TIME = 0.050 CONF.1-60



SIGMA R - DIAMOND SIGMA T - ASTERISK
SIGMA Z - CROSS TEMPERATURE - SQUARE

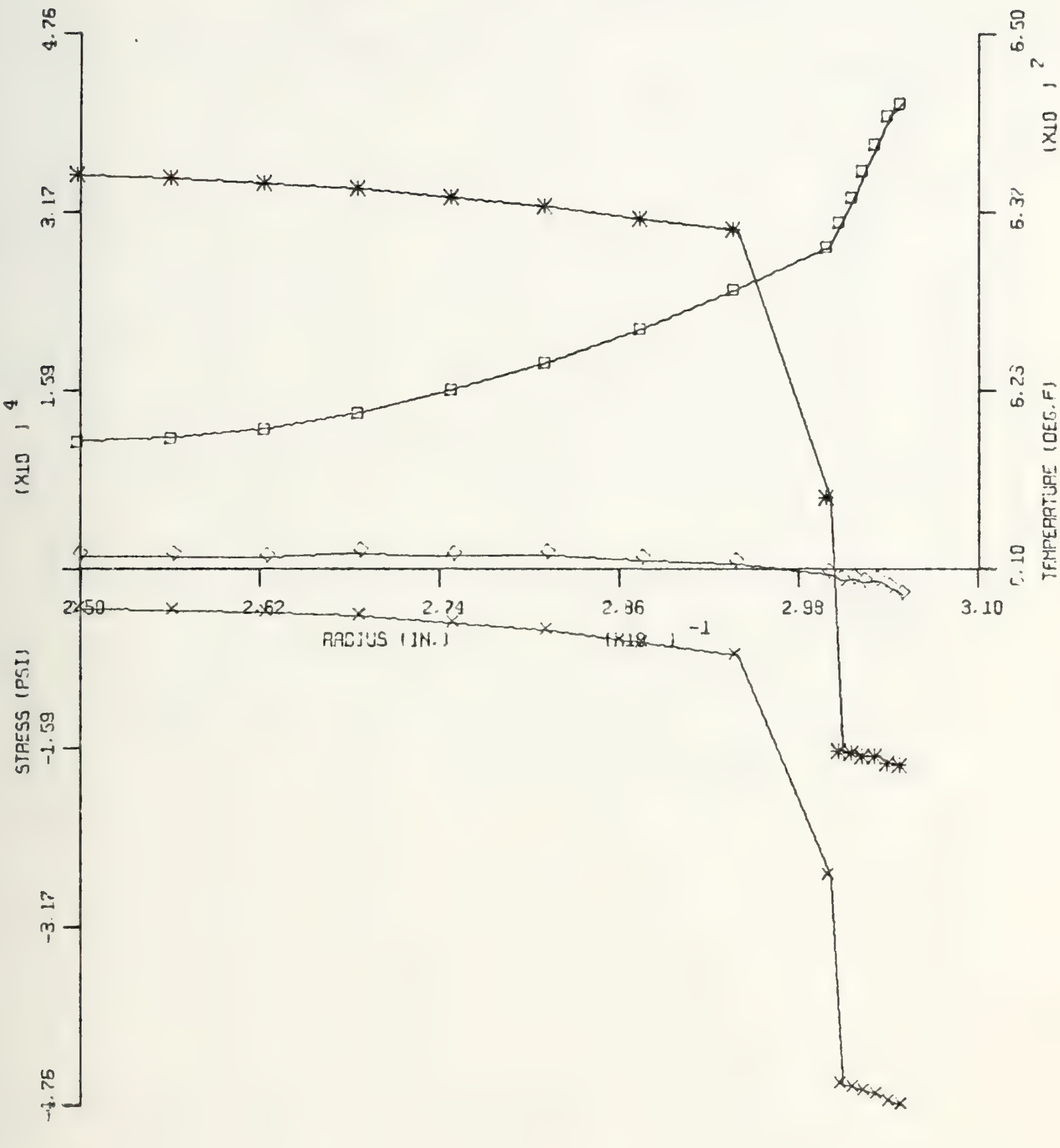
STRESS SCALE = 2.485X 10^4 PSI/INCH

TEMPERATURE SCALE = 13.3 DEGREES/INCH

RADIAL POSITION SCALE = 0.01 INCHES/INCH

FIGURE 4(a)

STRESS AND TEMPERATURE
VERSUS RADIAL POSITION
TIME = 0.100 CONF.1-60



SIGMA R - DIAMOND SIGMA T - ASTERISK
SIGMA Z - CROSS TEMPERATURE - SQUARE

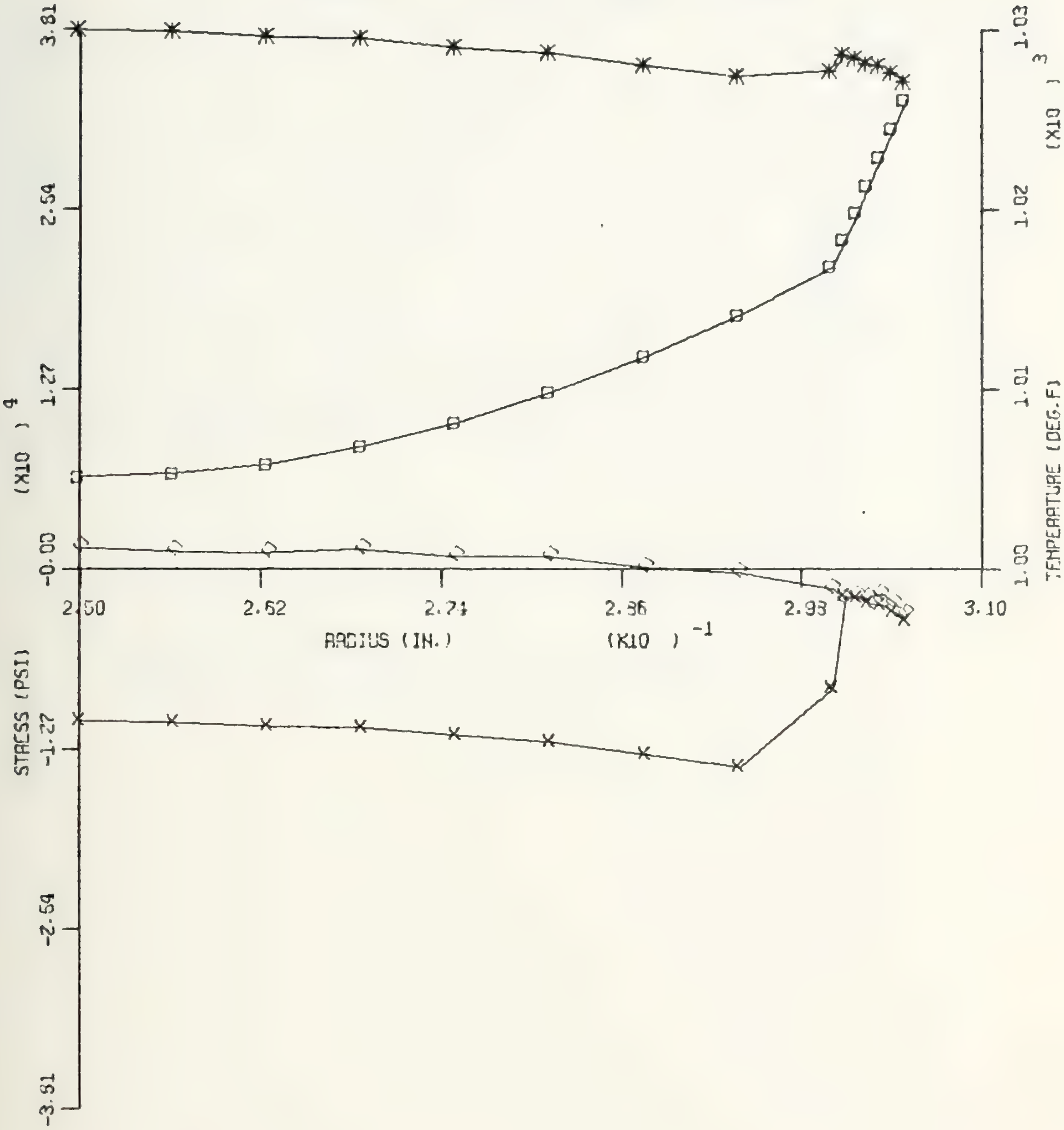
STRESS SCALE = 1.587X 10^4 PSI/INCH

TEMPERATURE SCALE = 13.3 DEGREES/INCH

RADIAL POSITION SCALE = 0.01 INCHES/INCH

FIGURE 4(b)

STRESS AND TEMPERATURE
VERSUS RADIAL POSITION
TIME = 0.200 CONF.1-60



SIGMA R - DIAMOND SIGMA T - ASTERISK
SIGMA Z - CROSS TEMPERATURE - SQUARE

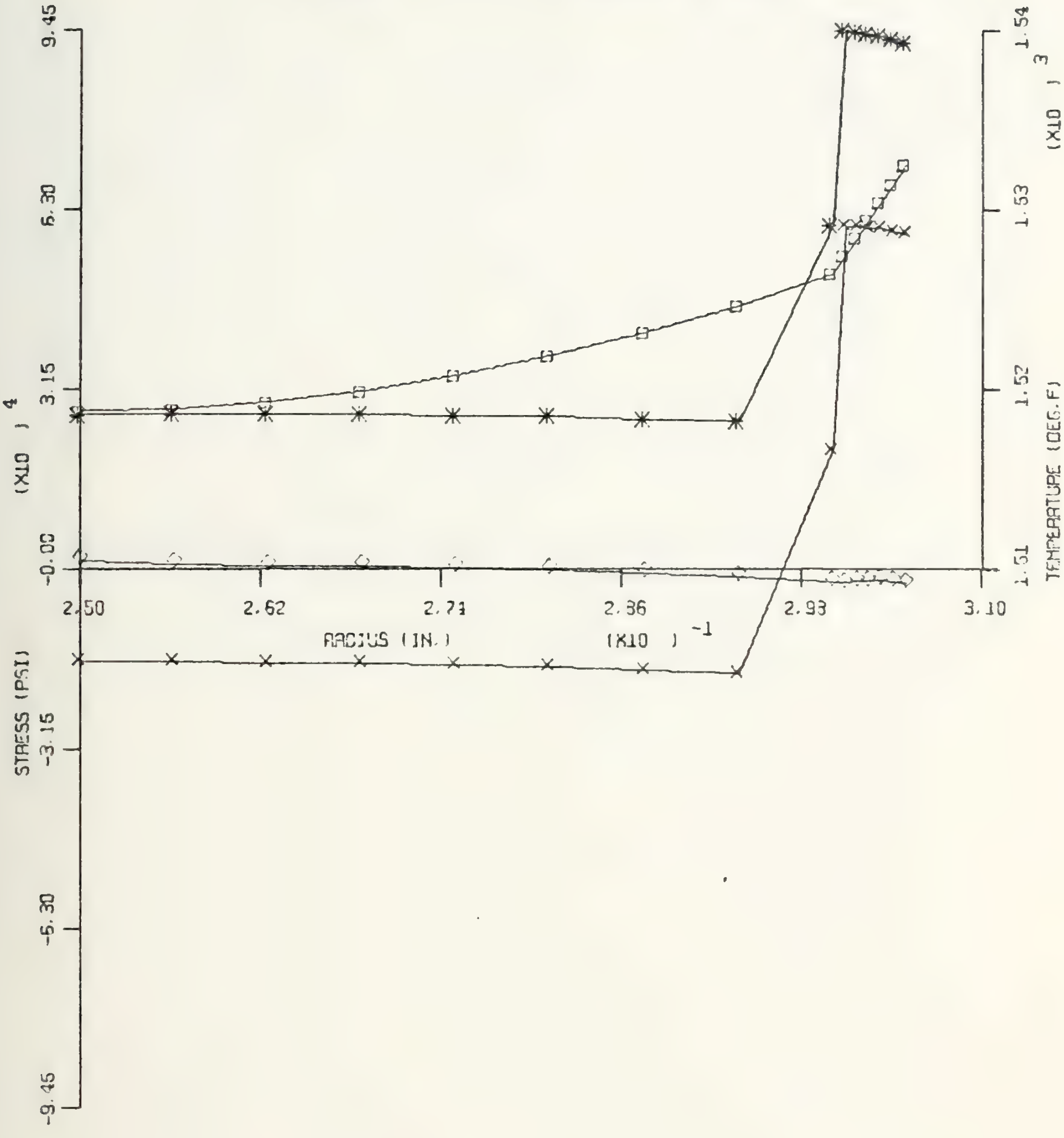
STRESS SCALE = 1.269X 10^{4} PSI/INCH

TEMPERATURE SCALE = 10.0 DEGREES/INCH

RADIAL POSITION SCALE = 0.01 INCHES/INCH

FIGURE 4(c)

STRESS AND TEMPERATURE
VERSUS RADIAL POSITION
TIME = 0.400 CONF.1-60



SIGMA R - DIAMOND SIGMA T - ASTERISK
SIGMA Z - CROSS TEMPERATURE - SQUARE

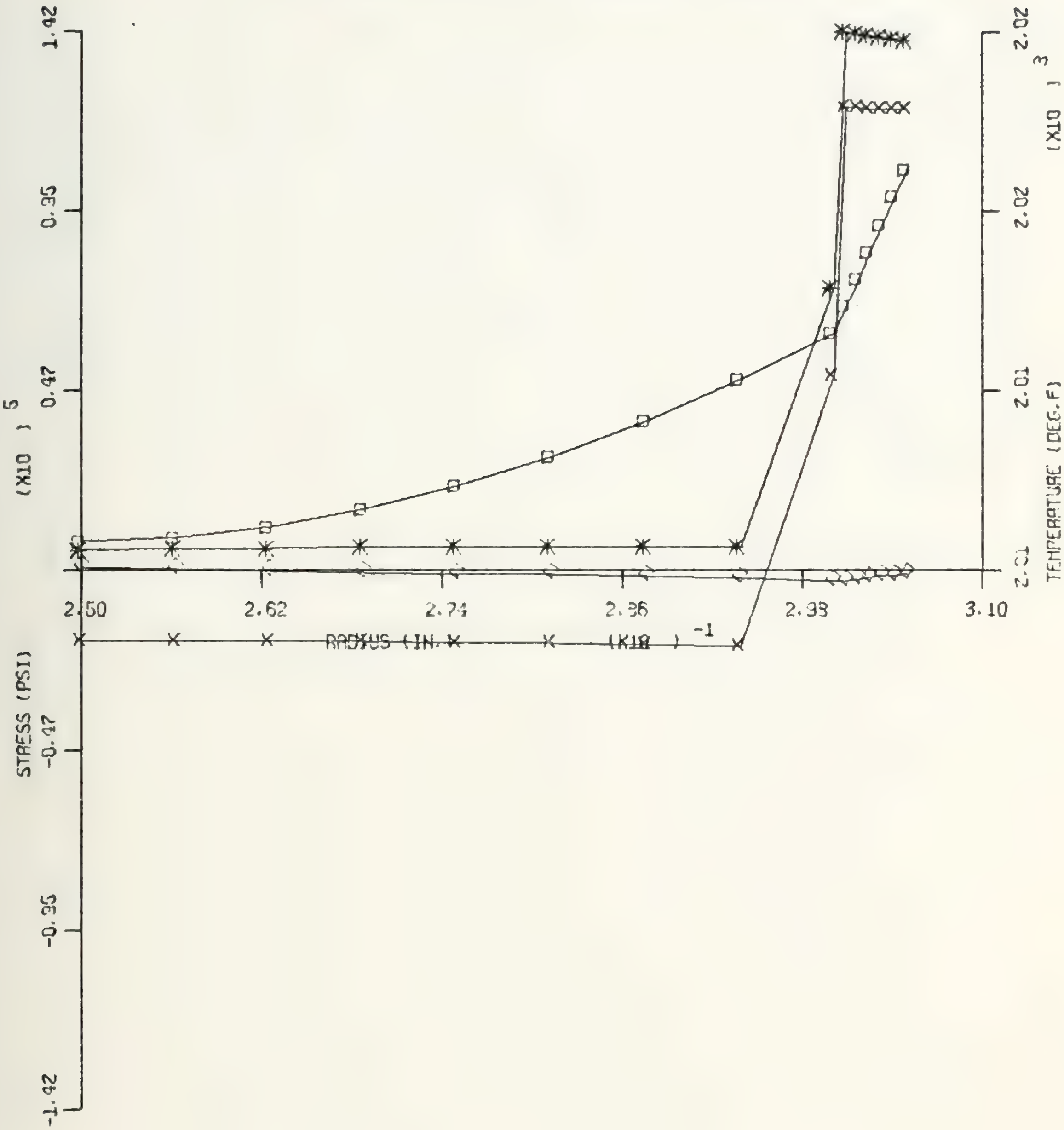
STRESS SCALE = 3.151X 10^{4} PSI/INCH

TEMPERATURE SCALE = 10.0 DEGREES/INCH

RADIAL POSITION SCALE = 0.01 INCHES/INCH

FIGURE 4(d)

STRESS AND TEMPERATURE
VERSUS RADIAL POSITION
TIME = 0.750 CONF.1-60



SIGMA R - DIAMOND SIGMA T - ASTERISK
SIGMA Z - CROSS TEMPERATURE - SQUARE

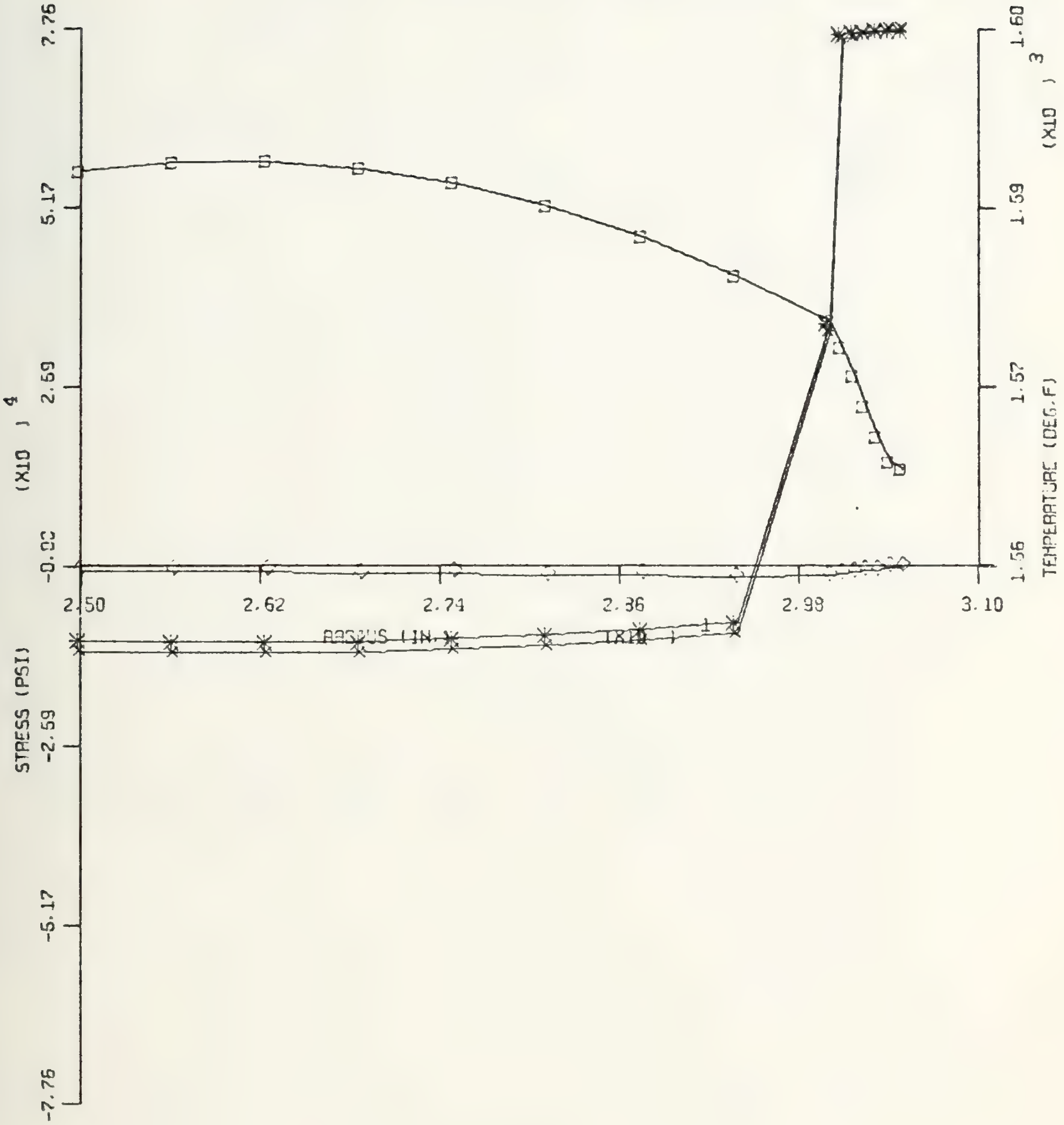
STRESS SCALE = 4.730X 10⁴ PSI/INCH

TEMPERATURE SCALE = 3.3 DEGREES/INCH

RADIAL POSITION SCALE = 0.01 INCHES/INCH

FIGURE 4(e)

STRESS AND TEMPERATURE
VERSUS RADIAL POSITION
TIME = 0.800 CONF.1-60



SIGMA R - DIAMOND SIGMA T - ASTERISK
SIGMA Z - CROSS TEMPERATURE - SQUARE

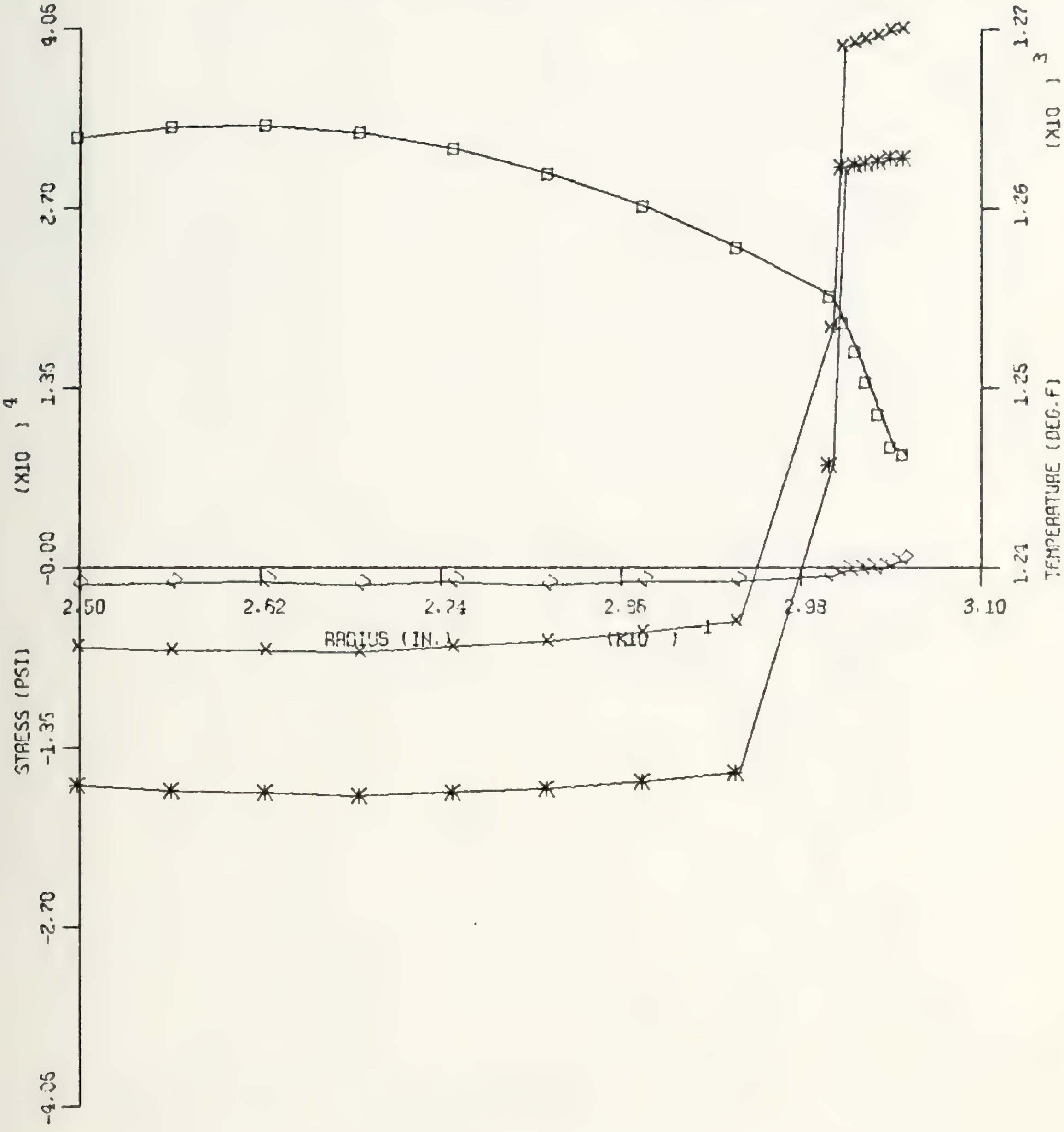
STRESS SCALE = 2.586X 10^3 PSI/INCH

TEMPERATURE SCALE = 13.3 DEGREES/INCH

RADIAL POSITION SCALE = 0.01 INCHES/INCH

FIGURE 4(f)

STRESS AND TEMPERATURE
VERSUS RADIAL POSITION
TIME = 0.850 CONF.1-60



SIGMA R - DIAMOND SIGMA T - ASTERISK
SIGMA Z - CROSS TEMPERATURE - SQUARE

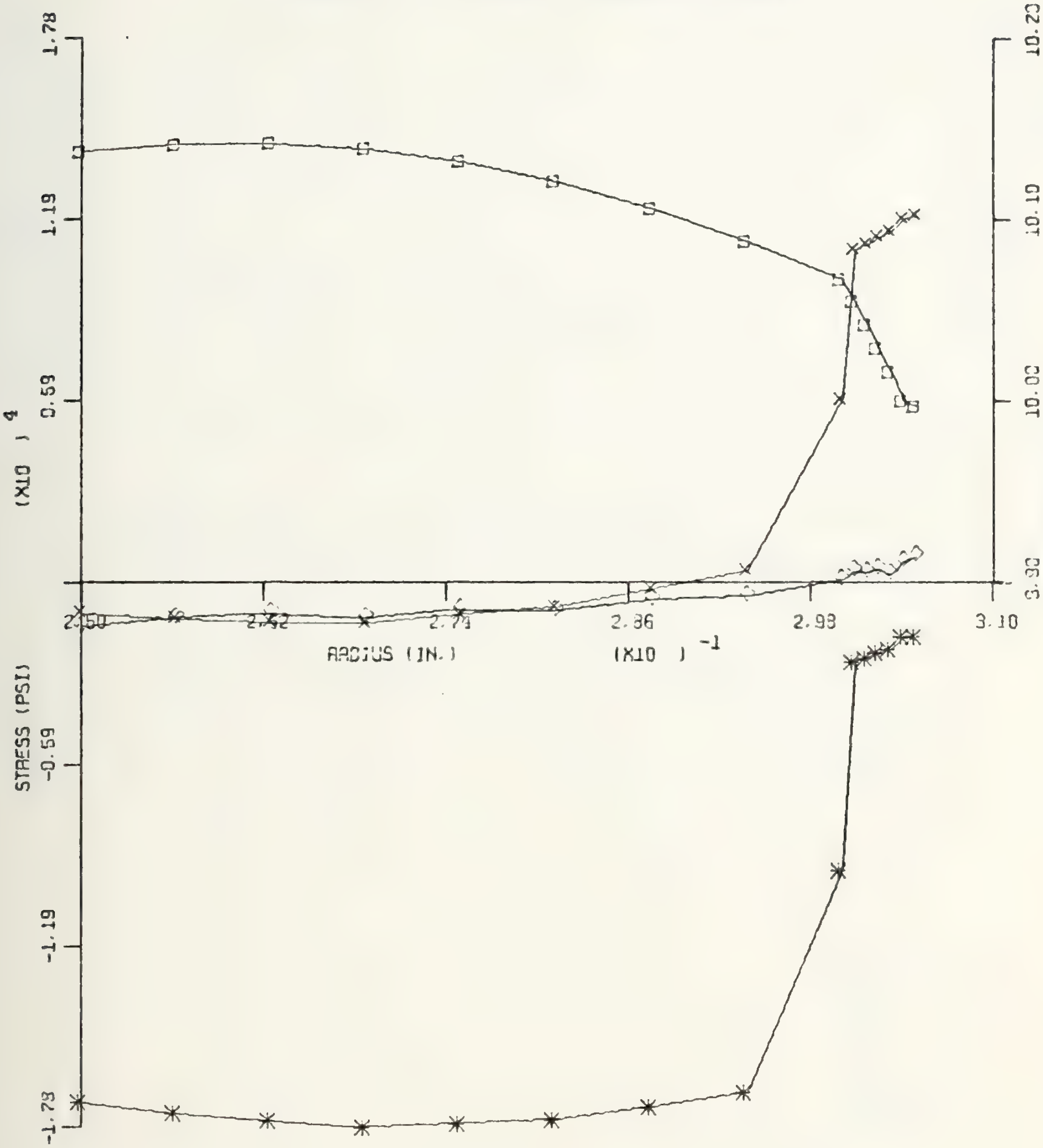
STRESS SCALE = 1.351×10^4 PSI/INCH

TEMPERATURE SCALE = 10.0 DEGREES/INCH

RADIAL POSITION SCALE = 0.01 INCHES/INCH

FIGURE 4(g)

STRESS AND TEMPERATURE
VERSUS RADIAL POSITION
TIME = 0.900 CONF.1-60



SIGMA R - DIAMOND   SIGMA T - ASTERISK
SIGMA Z - CROSS   TEMPERATURE - SQUARE

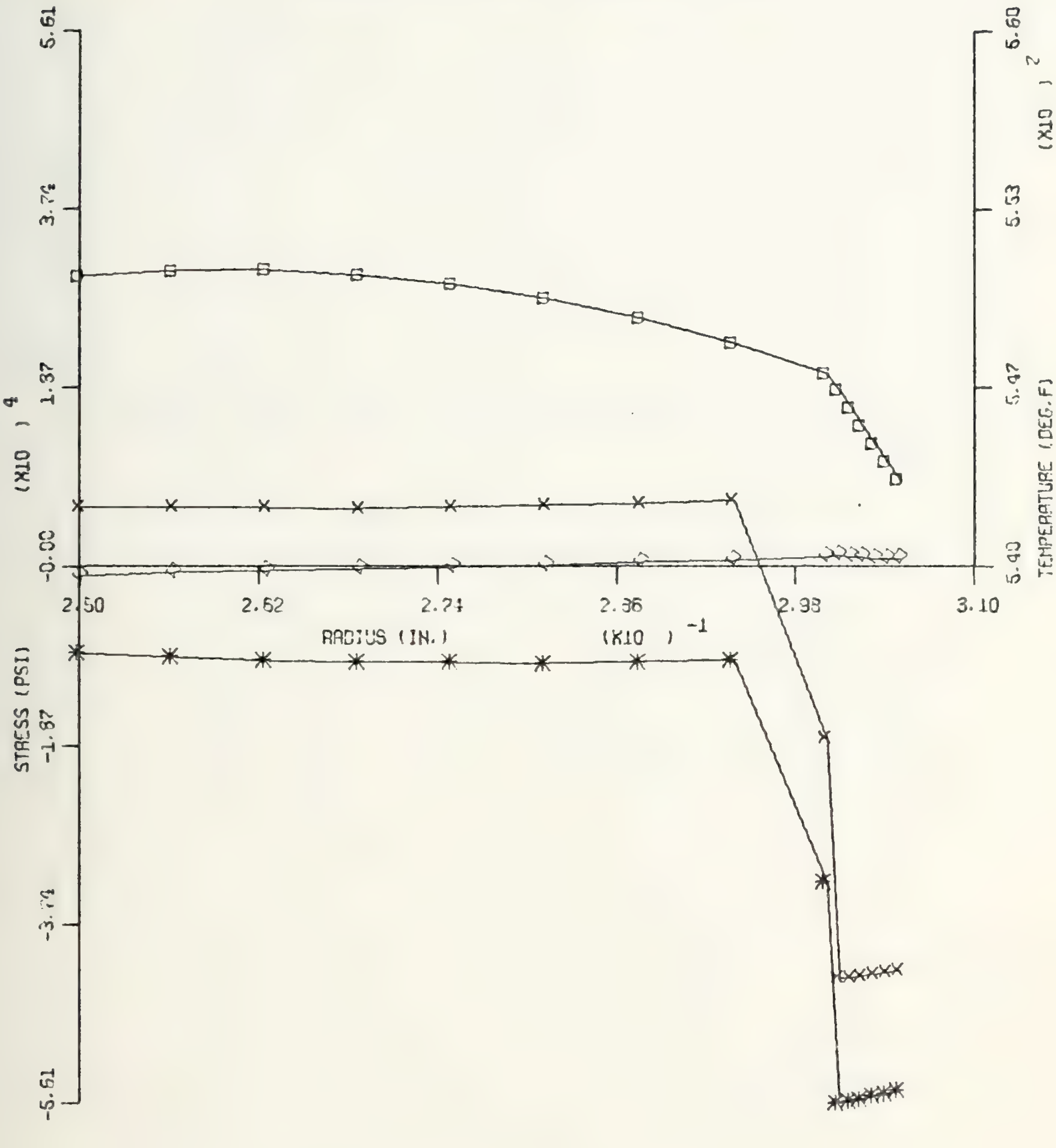
STRESS SCALE = 5.927×10^3 PSI/INCH

TEMPERATURE SCALE = 10.0 DEGREES/INCH

RADIAL POSITION SCALE = 0.01 INCHES/INCH

FIGURE 4(h)

STRESS AND TEMPERATURE
VERSUS RADIAL POSITION
TIME = 1.050 CONF.1-60



SIGMA R - DIAMOND SIGMA T - ASTERISK
SIGMA Z - CROSS TEMPERATURE - SQUARE

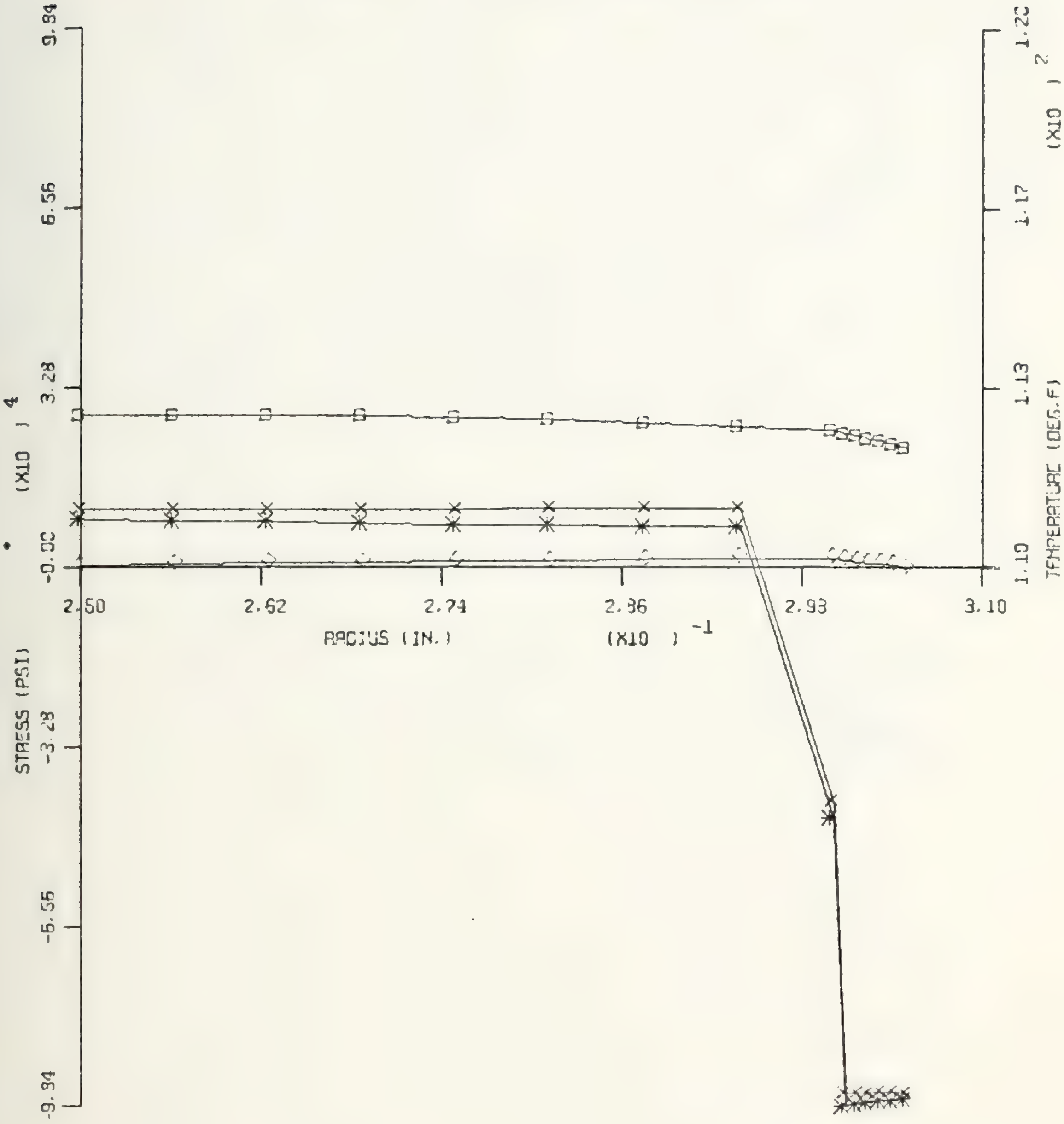
STRESS SCALE = 1.86SX 10^4 PSI/INCH

TEMPERATURE SCALE = 6.7 DEGREES/INCH

RADIAL POSITION SCALE = 0.01 INCHES/INCH

FIGURE 4(i)

STRESS AND TEMPERATURE
VERSUS RADIAL POSITION
TIME = 1.650 CONF.1-60



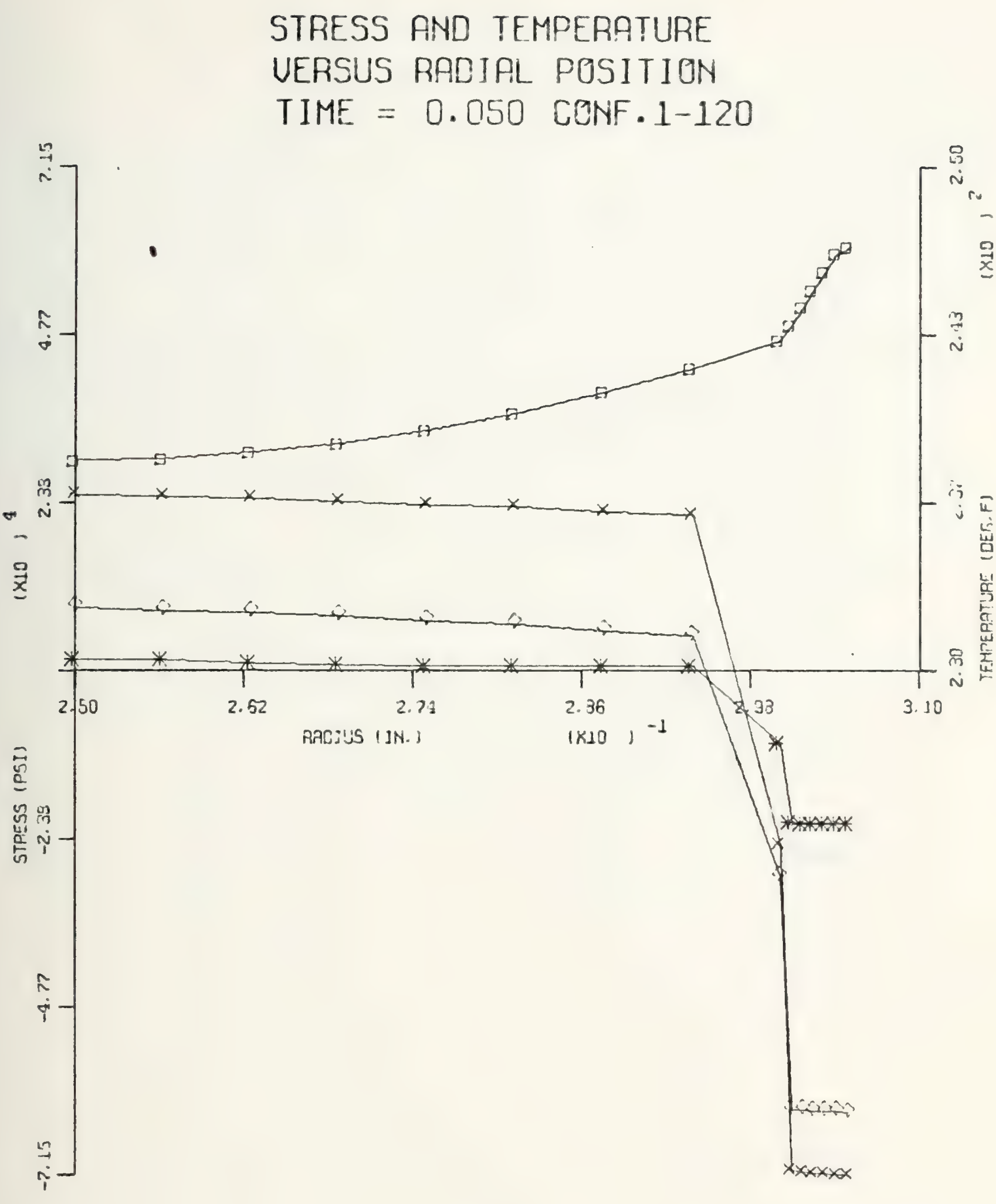
SIGMA R - DIAMOND SIGMA T - ASTERISK
SIGMA Z - CROSS TEMPERATURE - SQUARE

STRESS SCALE = 3.278X 10³ PSI/INCH

TEMPERATURE SCALE = 3.3 DEGREES/INCH

RADIAL POSITION SCALE = 0.01 INCHES/INCH

FIGURE 4(j)



SIGMA R - DIAMOND SIGMA T - ASTERISK
SIGMA Z - CROSS TEMPERATURE - SQUARE

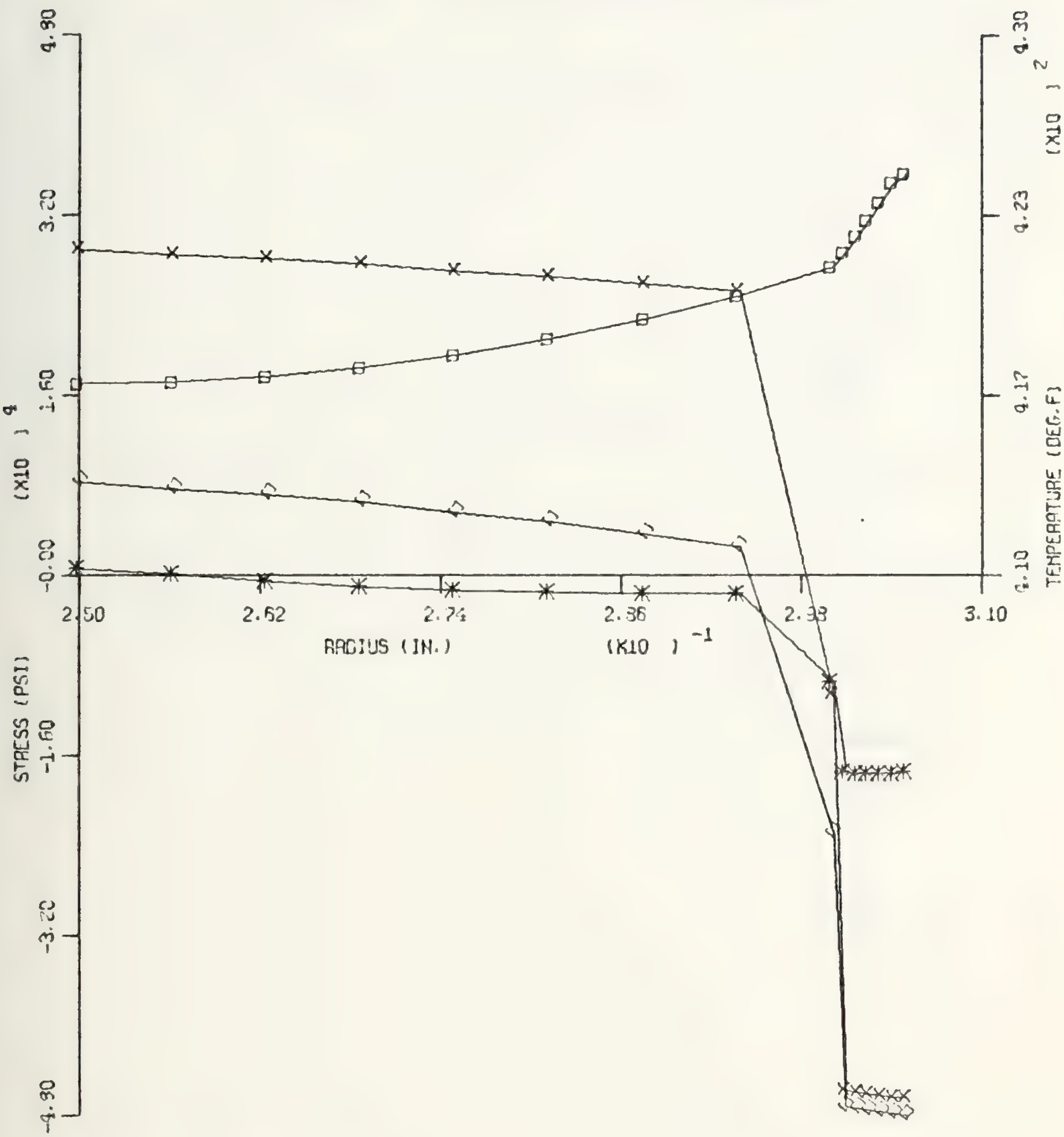
STRESS SCALE = 2.384X 10^3 PSI/INCH

TEMPERATURE SCALE = 6.7 DEGREES/INCH

RADIAL POSITION SCALE = 0.01 INCHES/INCH

FIGURE 5(a)

STRESS AND TEMPERATURE
VERSUS RADIAL POSITION
TIME = 0.100 CONF.1-120



SIGMA R - DIAMOND SIGMA T - ASTERISK
SIGMA Z - CROSS TEMPERATURE - SQUARE

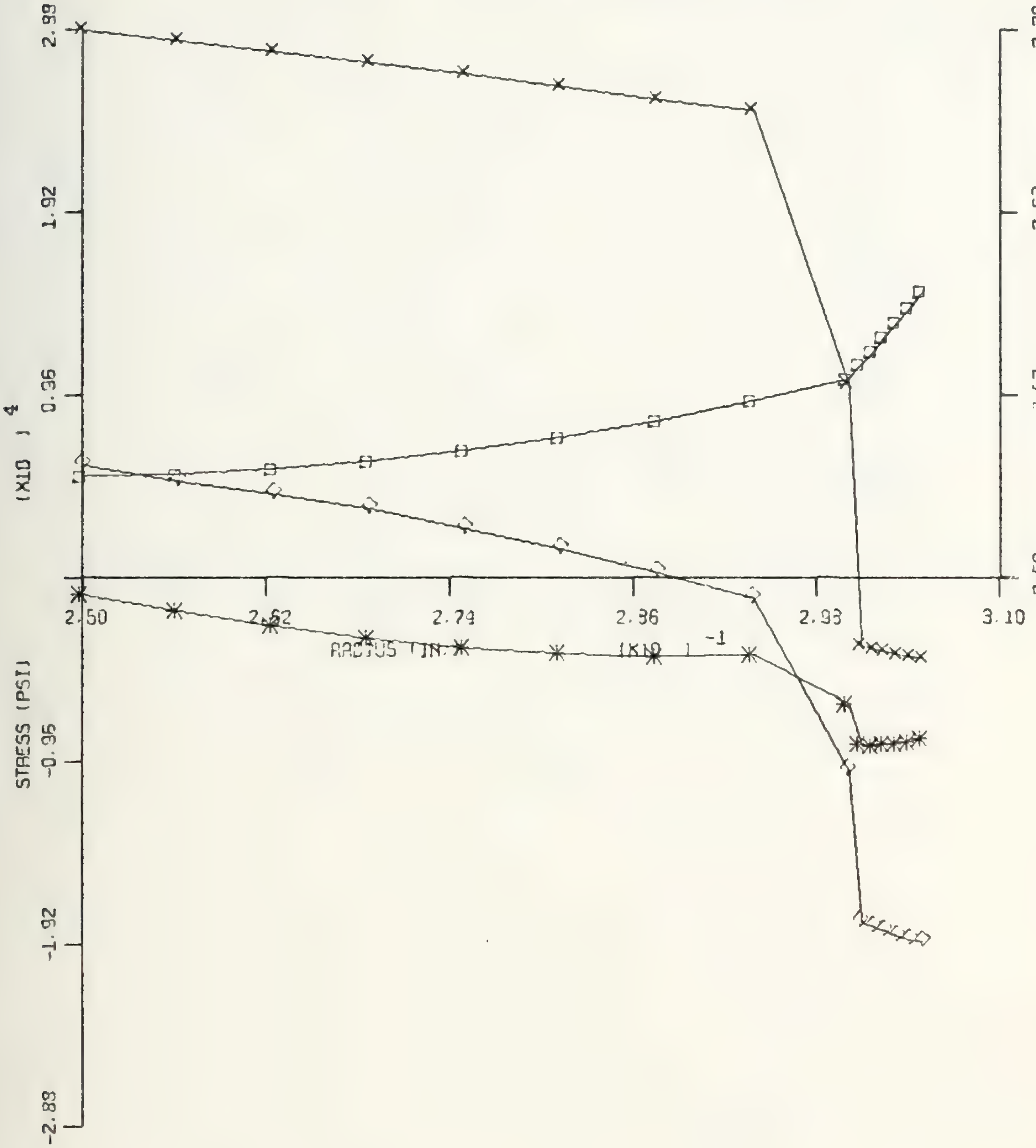
STRESS SCALE = 1.600X 10^4 PSI/INCH

TEMPERATURE SCALE = 6.7 DEGREES/INCH

RADIAL POSITION SCALE = 0.01 INCHES/INCH

FIGURE 5(b)

STRESS AND TEMPERATURE
VERSUS RADIAL POSITION
TIME = 0.200 CONF.1-120



SIGMA R - DIAMOND SIGMA T - ASTERISK
SIGMA Z - CROSS TEMPERATURE - SQUARE

STRESS SCALE = 3.597×10^{3} PSI/INCH

TEMPERATURE SCALE = 6.7 DEGREES/INCH

RADIAL POSITION SCALE = 0.01 INCHES/INCH

FIGURE 5(c)

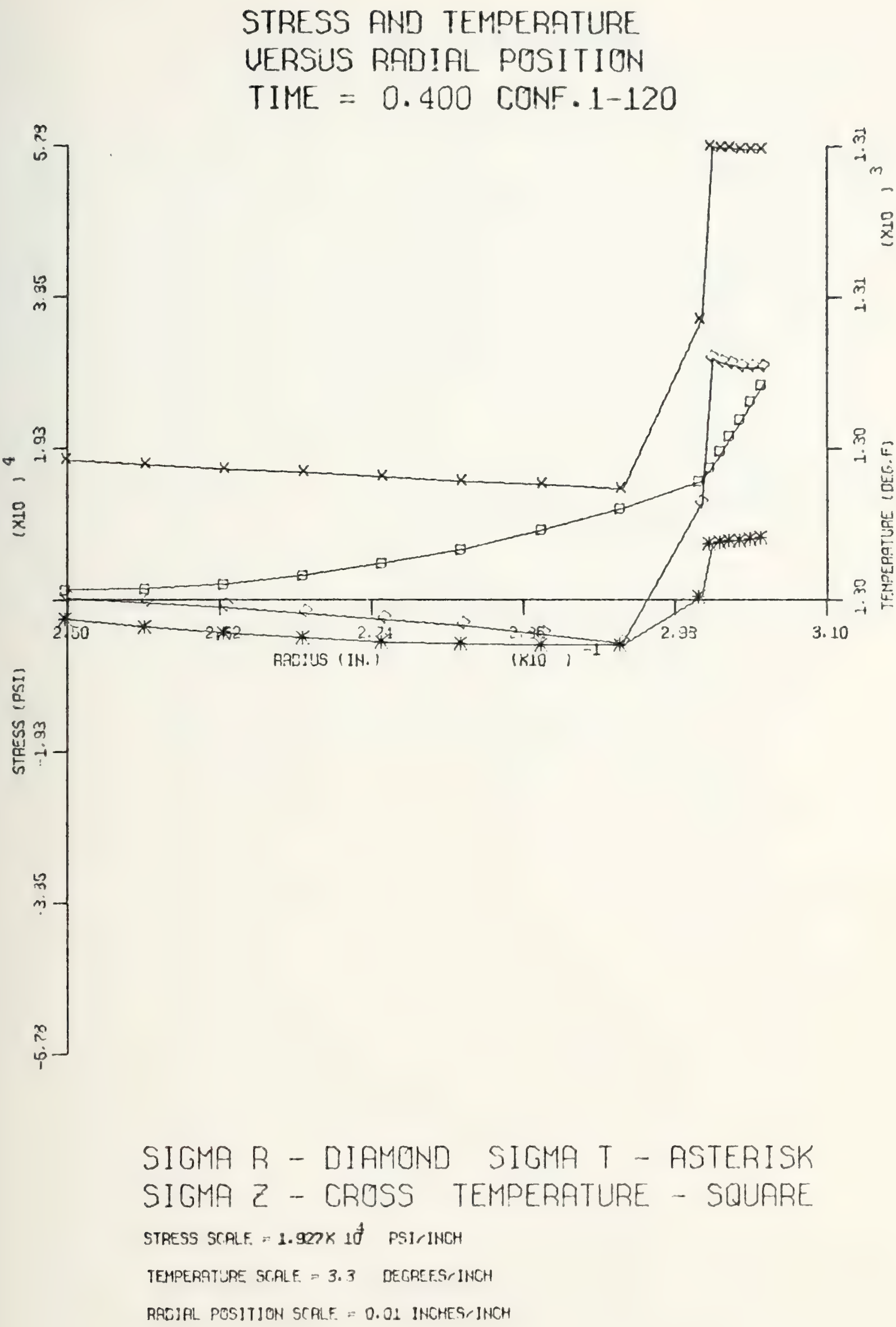
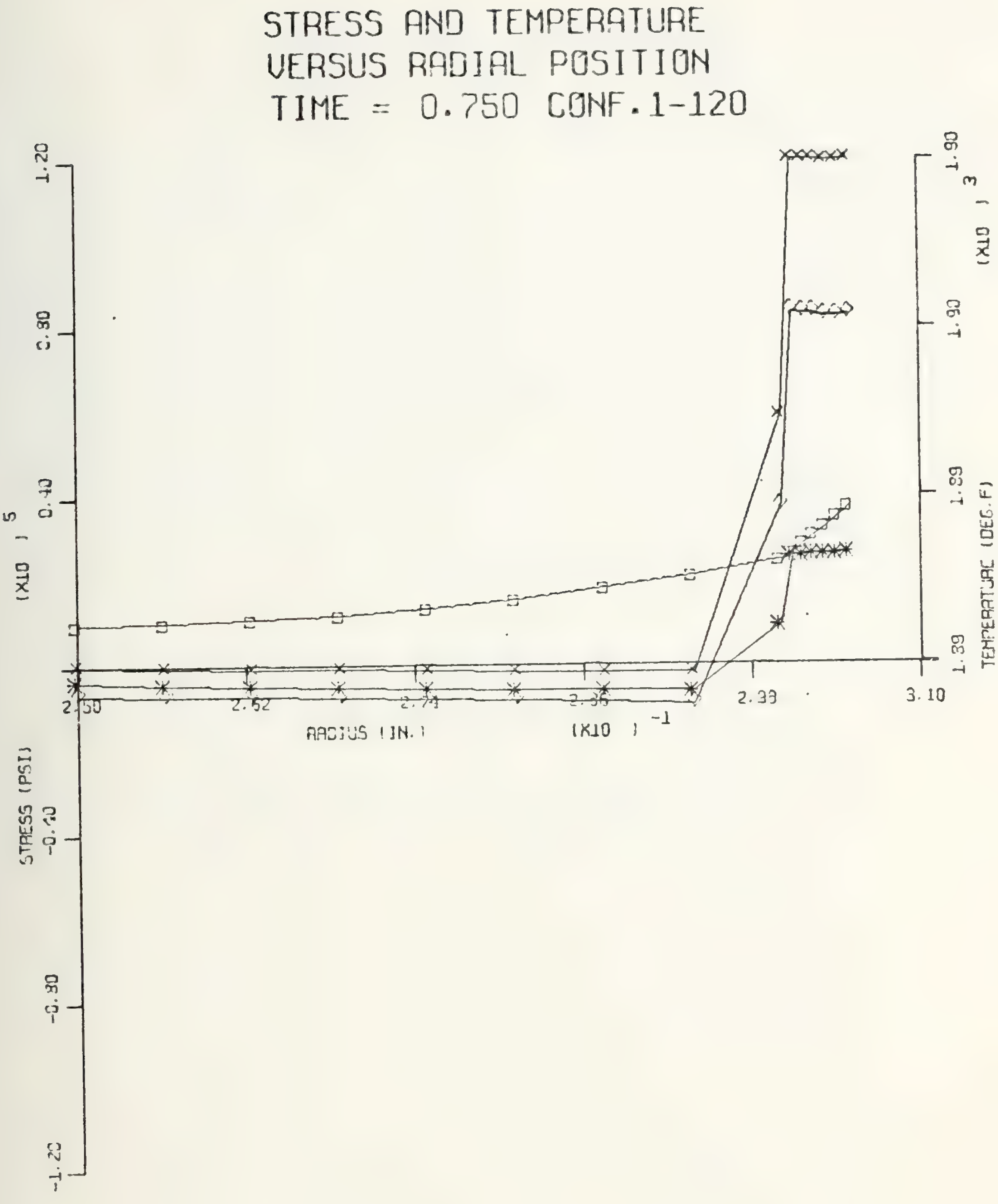

STRESS AND TEMPERATURE
VERSUS RADIAL POSITION
TIME = 0.400 CONF.1-120
5.78
3.85
1.93
-1.93
-3.85
-5.78
(X10) 4
STRESS (PSI)
1.31
1.31
1.30
1.30
(X10) 3
TEMPERATURE (DEG.F)
2.50
2.62
2.74
2.86
2.98
3.10
RADIUS (IN.)
(X10) -1
SIGMA R - DIAMOND SIGMA T - ASTERISK
SIGMA Z - CROSS TEMPERATURE - SQUARE
STRESS SCALE = 1.927X 10^4 PSI/INCH
TEMPERATURE SCALE = 3.3 DEGREES/INCH
RADIAL POSITION SCALE = 0.01 INCHES/INCH

FIGURE 5(d)



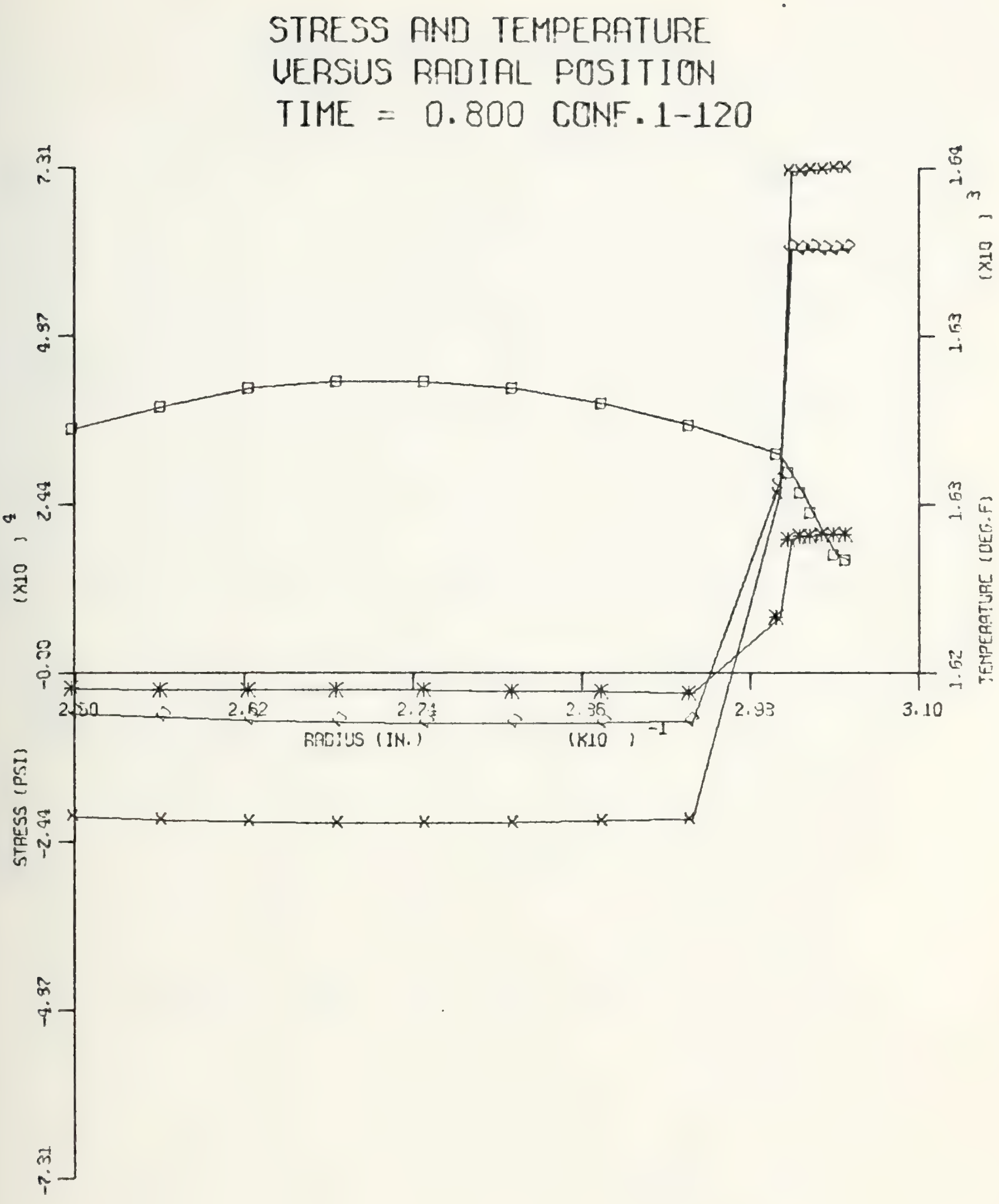
SIGMA R - DIAMOND SIGMA T - ASTERISK
SIGMA Z - CROSS TEMPERATURE - SQUARE

STRESS SCALE = 4.016×10^{4} PSI/INCH

TEMPERATURE SCALE = 3.3 DEGREES/INCH

RADIAL POSITION SCALE = 0.01 INCHES/INCH

FIGURE 5(e)



SIGMA R - DIAMOND SIGMA T - ASTERISK
SIGMA Z - CROSS TEMPERATURE - SQUARE

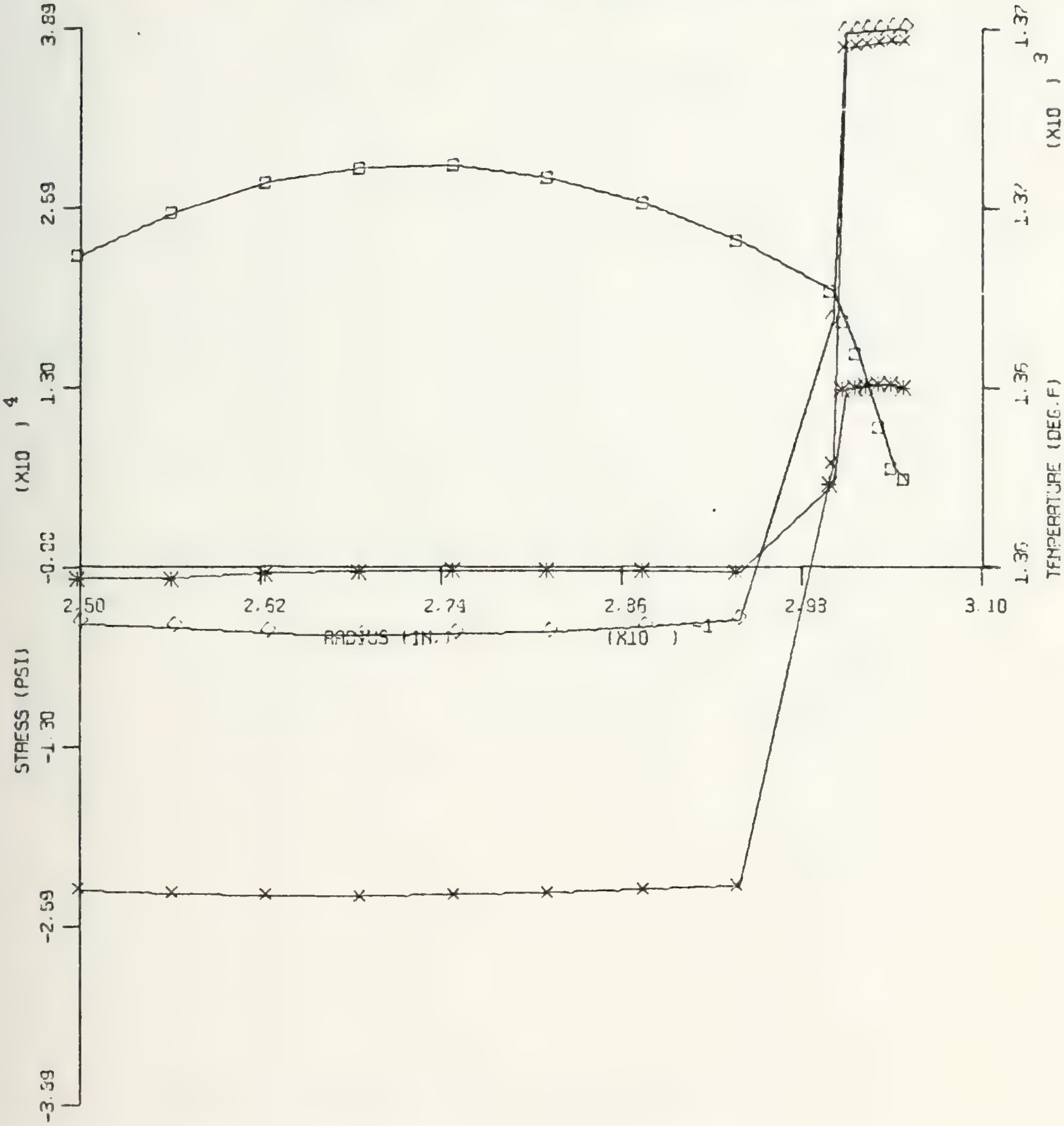
STRESS SCALE = 2.437X 10^4 PSI/INCH

TEMPERATURE SCALE = 6.7 DEGREES/INCH

RADIAL POSITION SCALE = 0.01 INCHES/INCH

FIGURE 5(f)

STRESS AND TEMPERATURE
VERSUS RADIAL POSITION
TIME = 0.850 CONF.1-120



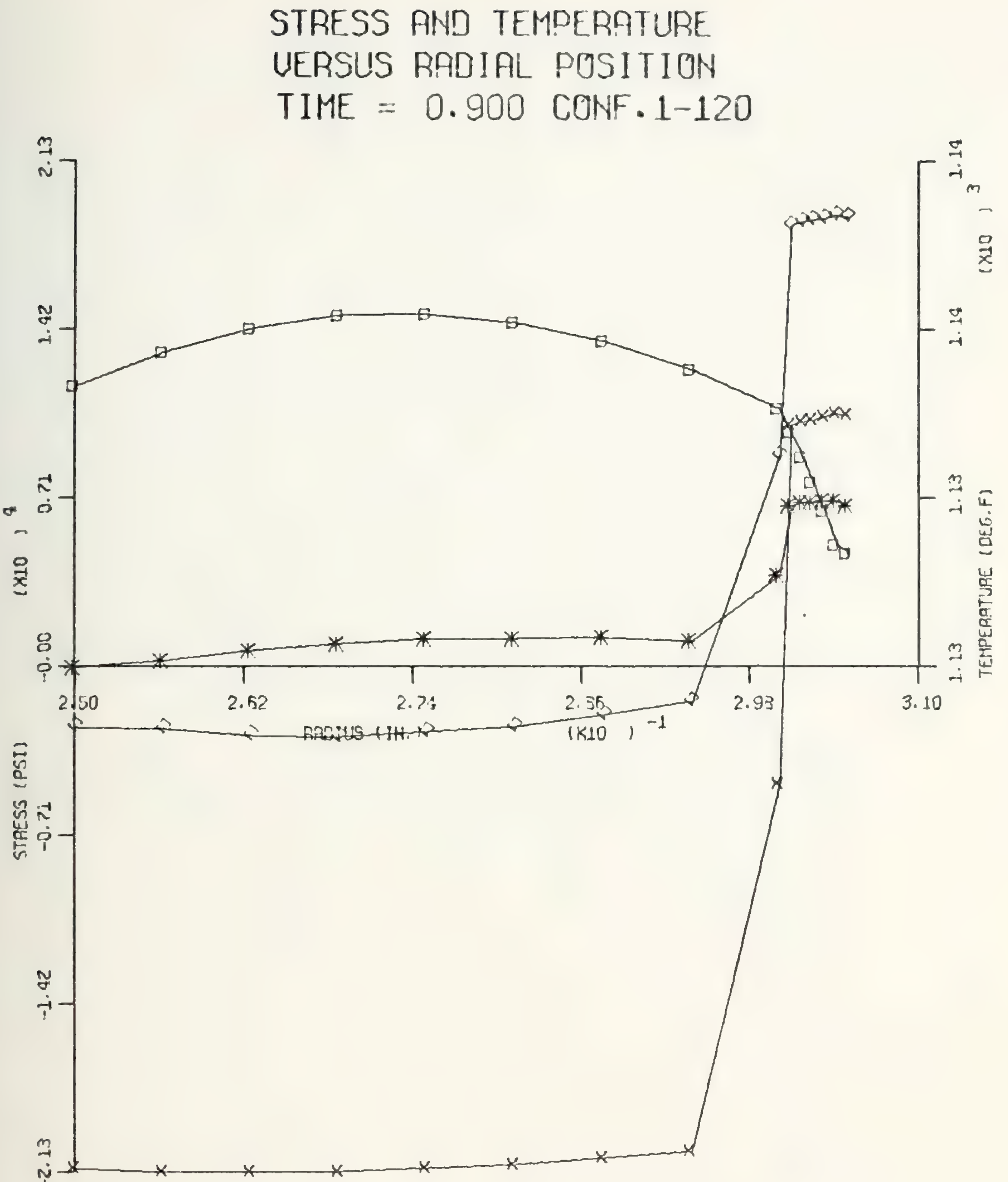
SIGMA R - DIAMOND SIGMA T - ASTERISK
SIGMA Z - CROSS TEMPERATURE - SQUARE

STRESS SCALE = 1.235X 10^4 PSI/INCH

TEMPERATURE SCALE = 3.3 DEGREES/INCH

RADIAL POSITION SCALE = 0.01 INCHES/INCH

FIGURE 5(g)



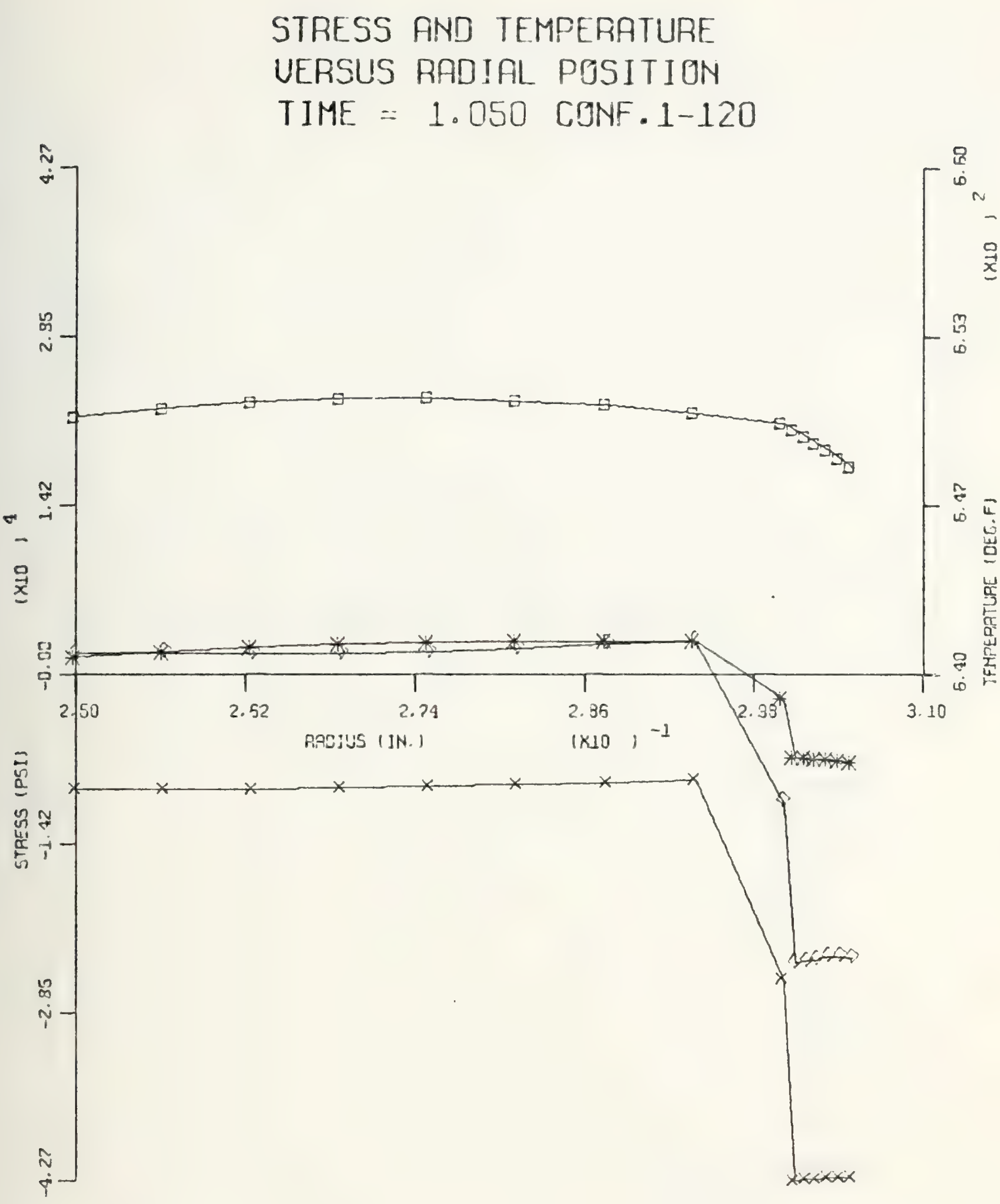
SIGMA R - DIAMOND SIGMA T - ASTERISK
SIGMA Z - CROSS TEMPERATURE - SQUARE

STRESS SCALE = 7.093X 10^3 PSI/INCH

TEMPERATURE SCALE = 3.3 DEGREES/INCH

RADIAL POSITION SCALE = 0.01 INCHES/INCH

FIGURE 5(h)



SIGMA R - DIAMOND SIGMA T - ASTERISK
SIGMA Z - CROSS TEMPERATURE - SQUARE

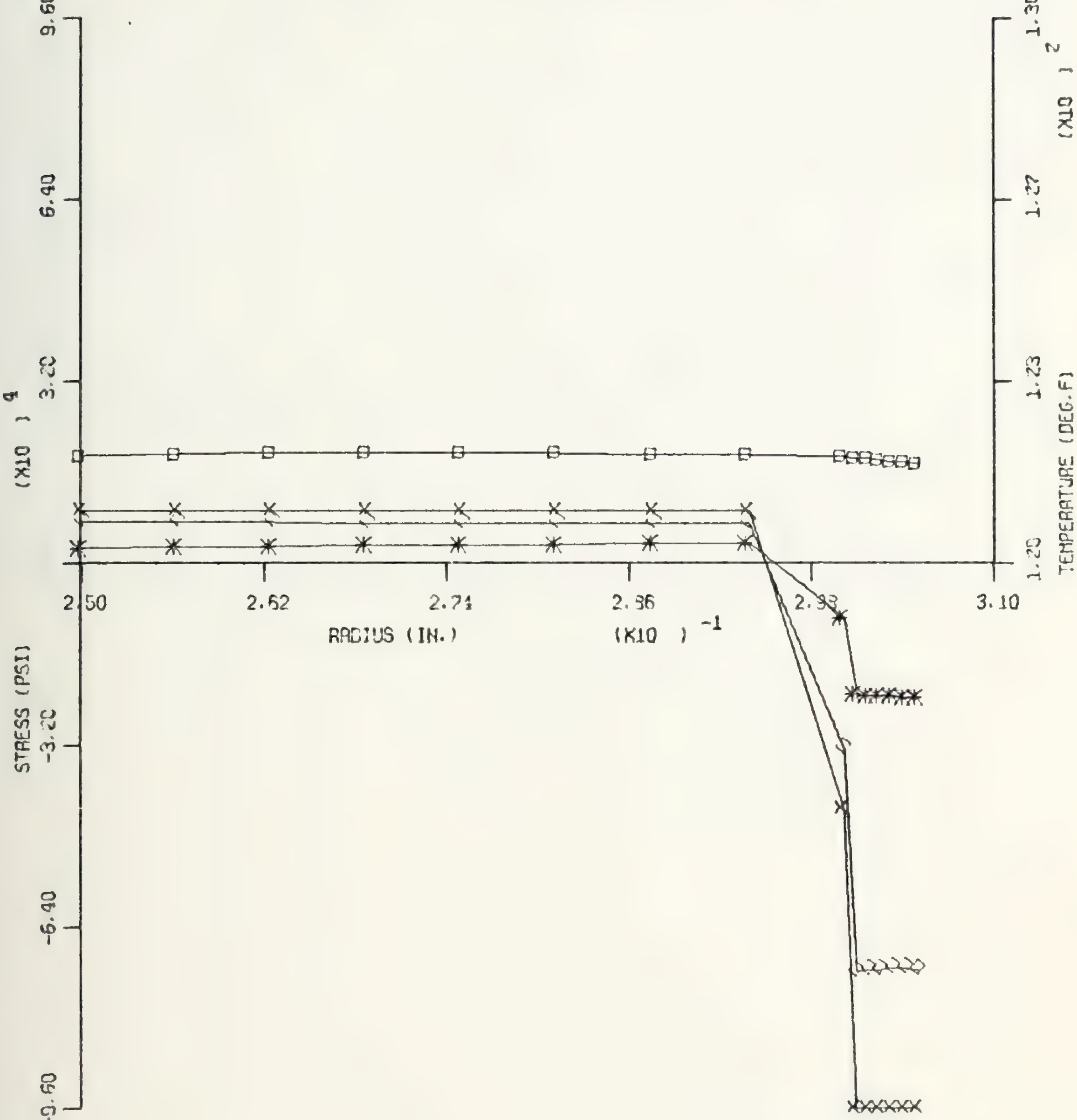
STRESS SCALE = 1.423X 10^4 PSI/INCH

TEMPERATURE SCALE = 6.7 DEGREES/INCH

RADIAL POSITION SCALE = 0.01 INCHES/INCH

FIGURE 5(i)

STRESS AND TEMPERATURE
VERSUS RADIAL POSITION
TIME = 1.650 CONF.1-120



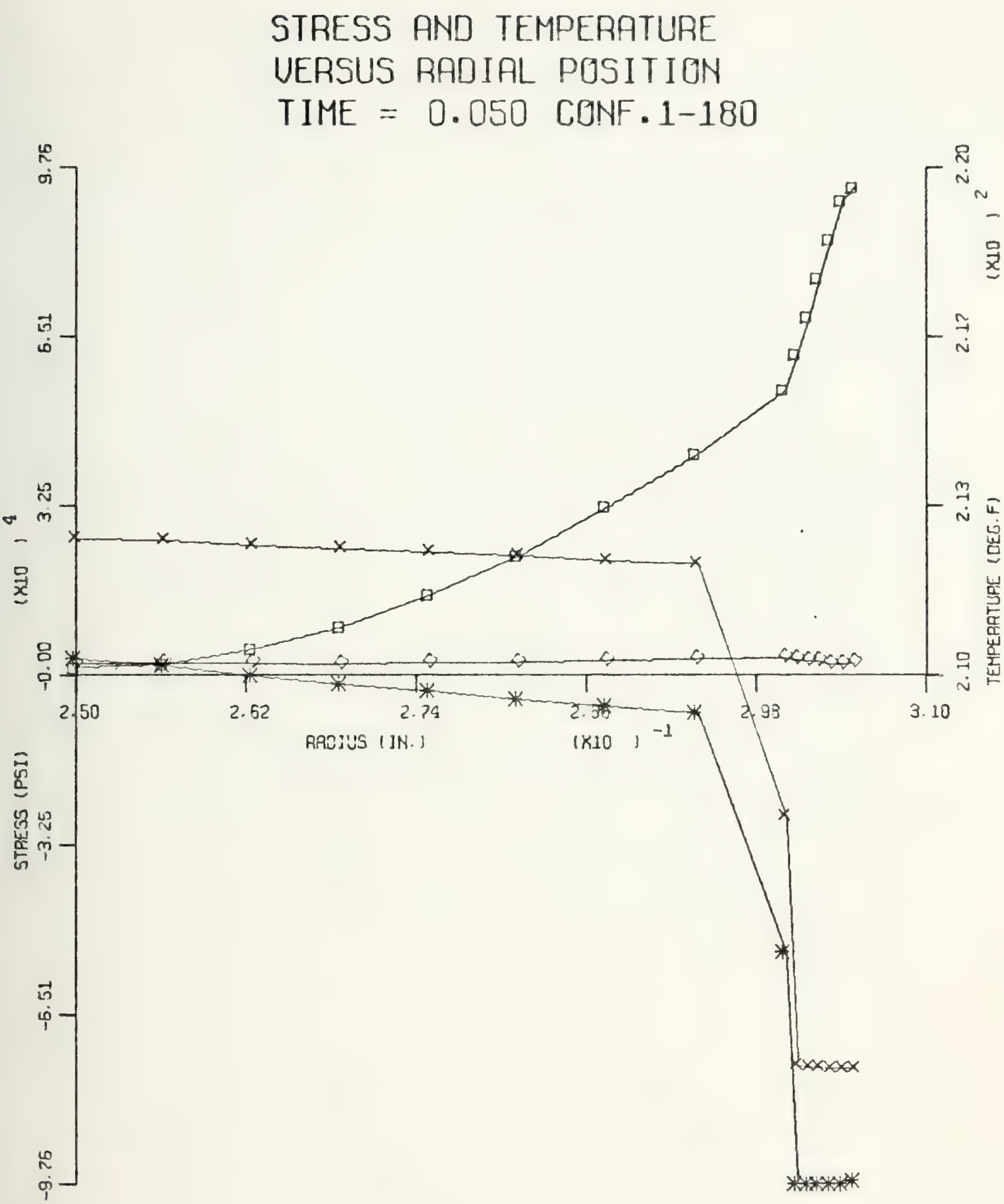
SIGMA R - DIAMOND SIGMA T - ASTERISK
SIGMA Z - CROSS TEMPERATURE - SQUARE

STRESS SCALE = 3.193X 10^4 PSI/INCH

TEMPERATURE SCALE = 3.3 DEGREES/INCH

RADIAL POSITION SCALE = 0.01 INCHES/INCH

FIGURE 5(j)



SIGMA R - DIAMOND SIGMA T - ASTERISK
SIGMA Z - CROSS TEMPERATURE - SQUARE

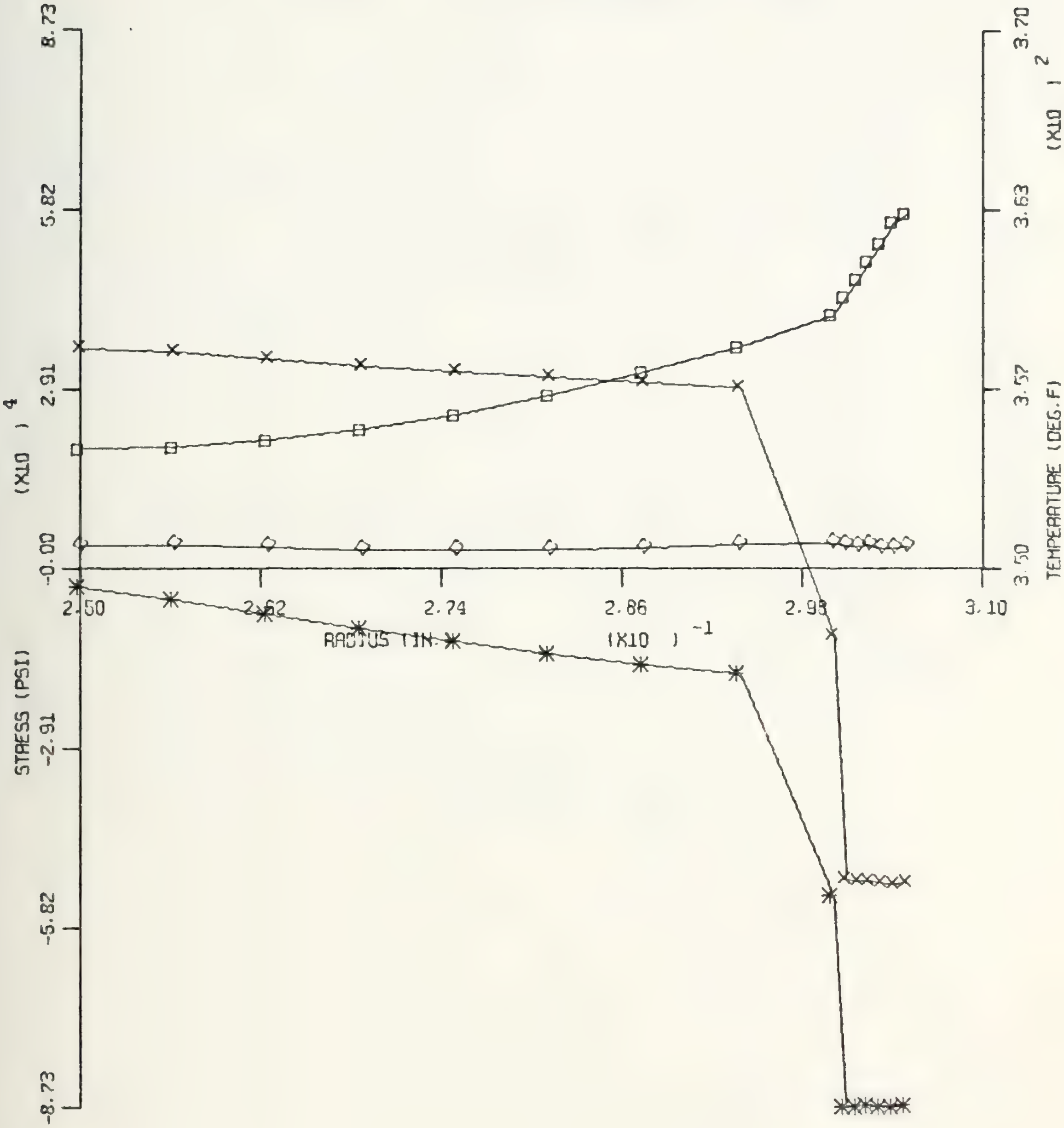
STRESS SCALE = 3.254X 10 4 PSI/INCH

TEMPERATURE SCALE = 3.3 DEGREES/INCH

RADIAL POSITION SCALE = 0.01 INCHES/INCH

FIGURE 6(a)

STRESS AND TEMPERATURE
VERSUS RADIAL POSITION
TIME = 0.100 CONF.1-180



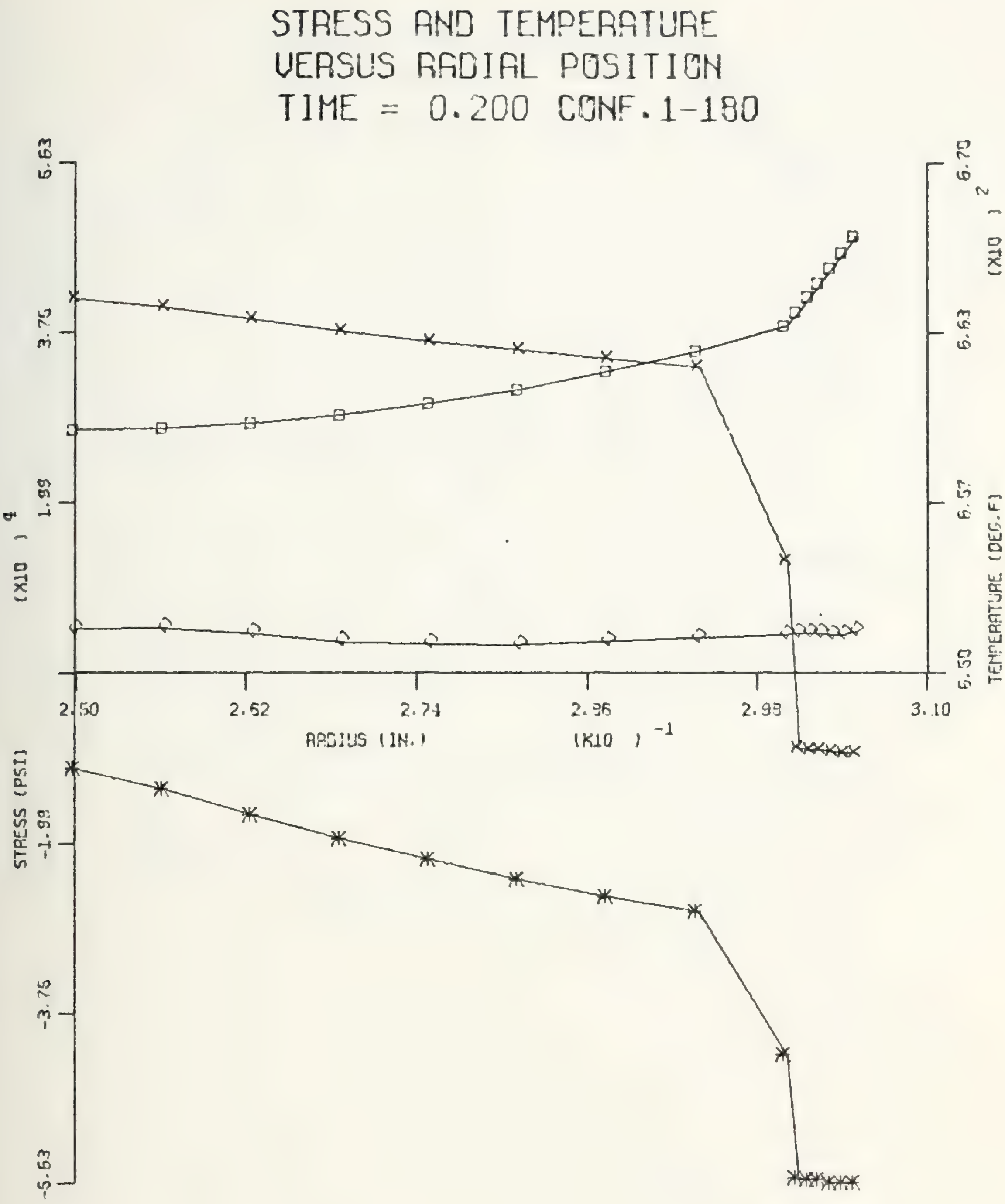
SIGMA R - DIAMOND SIGMA T - ASTERISK
SIGMA Z - CROSS TEMPERATURE - SQUARE

STRESS SCALE = 2.909X 10³ PSI/INCH

TEMPERATURE SCALE = 6.7 DEGREES/INCH

RADIAL POSITION SCALE = 0.01 INCHES/INCH

FIGURE 6(b)



SIGMA R - DIAMOND SIGMA T - ASTERISK
SIGMA Z - CROSS TEMPERATURE - SQUARE

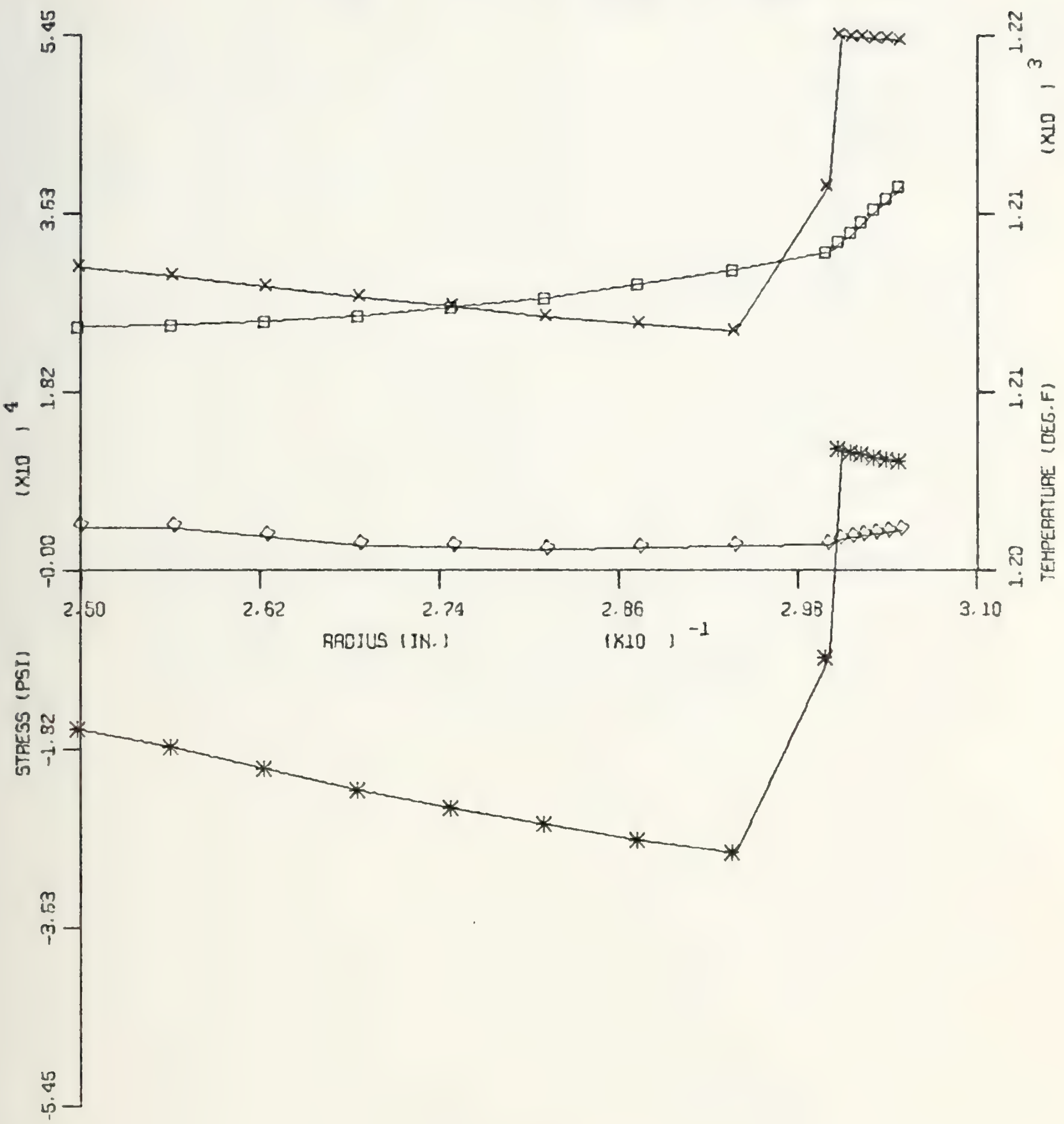
STRESS SCALE = 1.877X 10^4 PSI/INCH

TEMPERATURE SCALE = 6.7 DEGREES/INCH

RADIAL POSITION SCALE = 0.01 INCHES/INCH

FIGURE 6(c)

STRESS AND TEMPERATURE
VERSUS RADIAL POSITION
TIME = 0.400 CONF.1-180



SIGMA R - DIAMOND SIGMA T - ASTERISK
SIGMA Z - CROSS TEMPERATURE - SQUARE

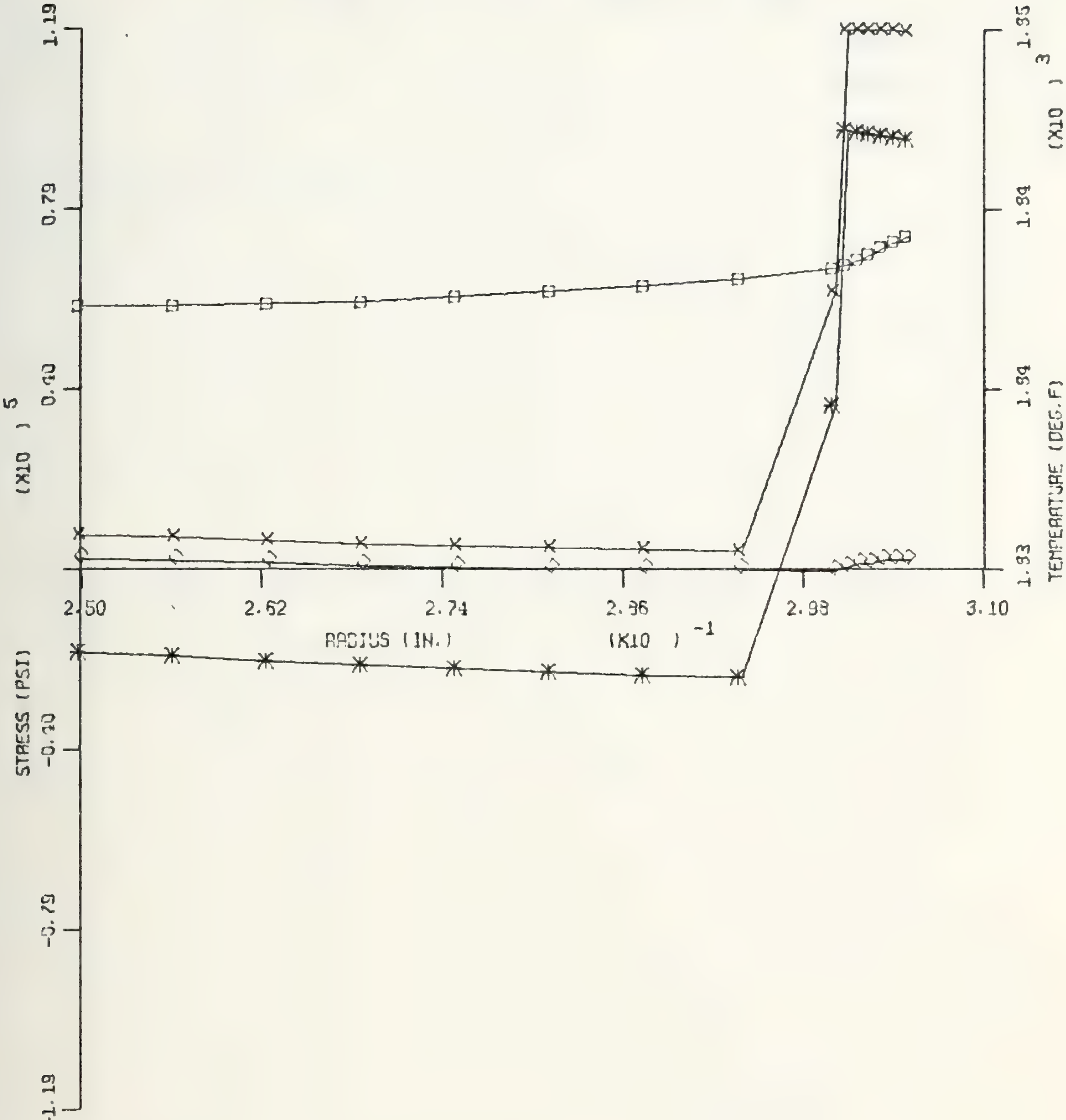
STRESS SCALE = 1.815X 10^4 PSI/INCH

TEMPERATURE SCALE = 6.7 DEGREES/INCH

RADIAL POSITION SCALE = 0.01 INCHES/INCH

FIGURE 6(d)

STRESS AND TEMPERATURE
VERSUS RADIAL POSITION
TIME = 0.750 CONF.1-180



SIGMA R - DIAMOND SIGMA T - ASTERISK
SIGMA Z - CROSS TEMPERATURE - SQUARE

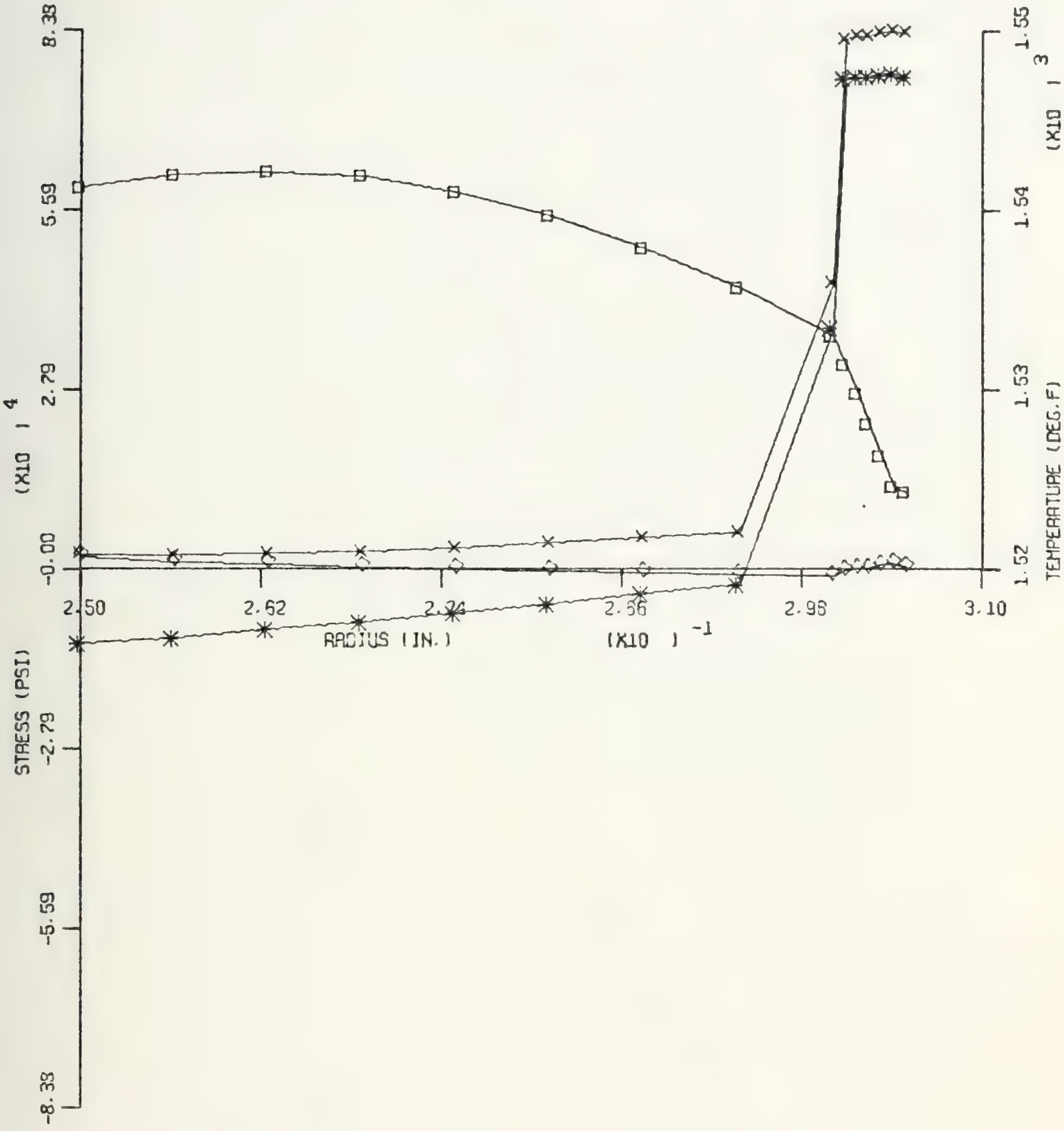
STRESS SCALE = 3.957X 10^4 PSI/INCH

TEMPERATURE SCALE = 6.7 DEGREES/INCH

RADIAL POSITION SCALE = 0.01 INCHES/INCH

FIGURE 6(e)

STRESS AND TEMPERATURE
VERSUS RADIAL POSITION
TIME = 0.800 CONF.1-180



SIGMA R - DIAMOND SIGMA T - ASTERISK
SIGMA Z - CROSS TEMPERATURE - SQUARE

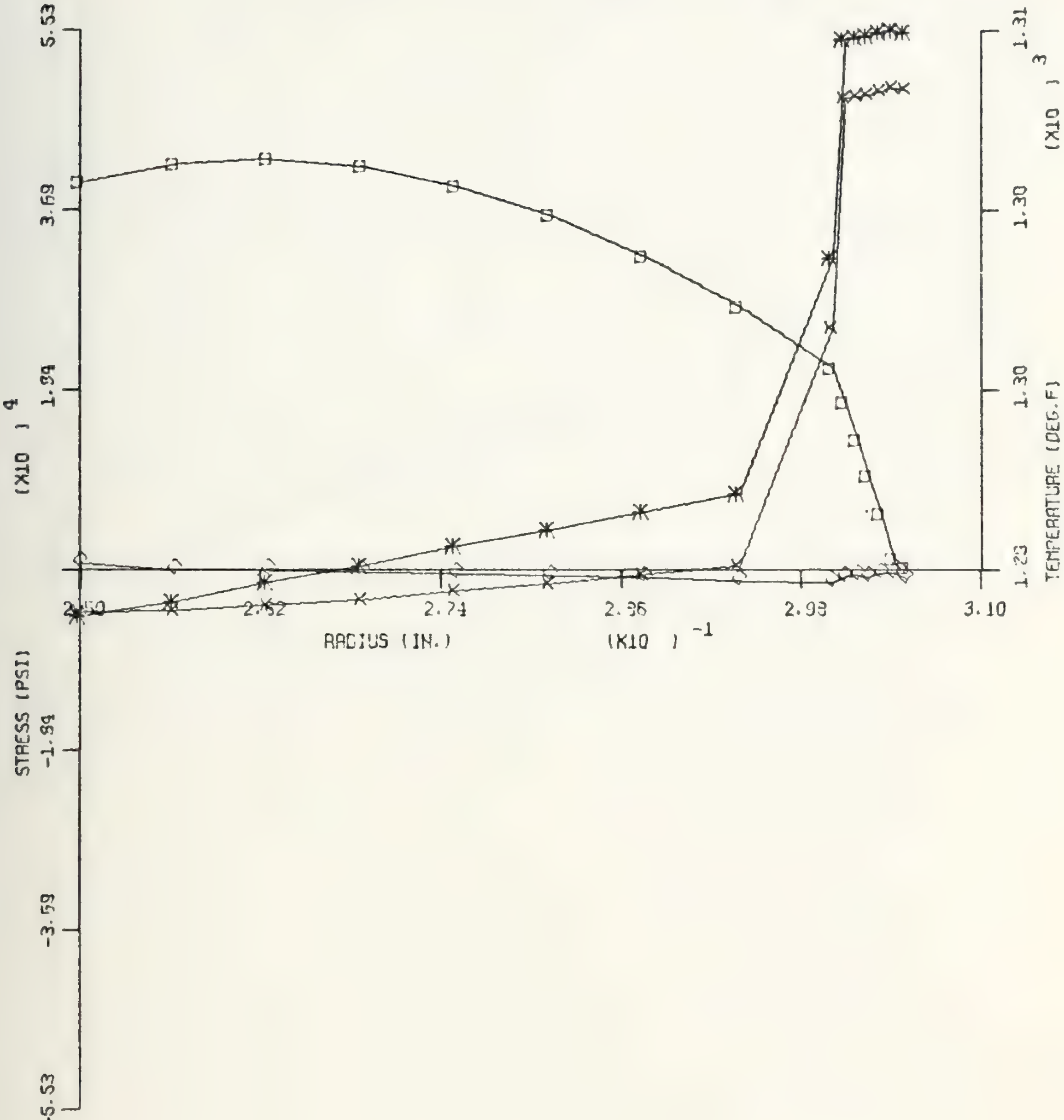
STRESS SCALE = 2.793X 10^{3} PSI/INCH

TEMPERATURE SCALE = 10.0 DEGREES/INCH

RADIAL POSITION SCALE = 0.01 INCHES/INCH

FIGURE 6(f)

STRESS AND TEMPERATURE
VERSUS RADIAL POSITION
TIME = 0.850 CONF.1-180



SIGMA R - DIAMOND SIGMA T - ASTERISK
SIGMA Z - CROSS TEMPERATURE - SQUARE

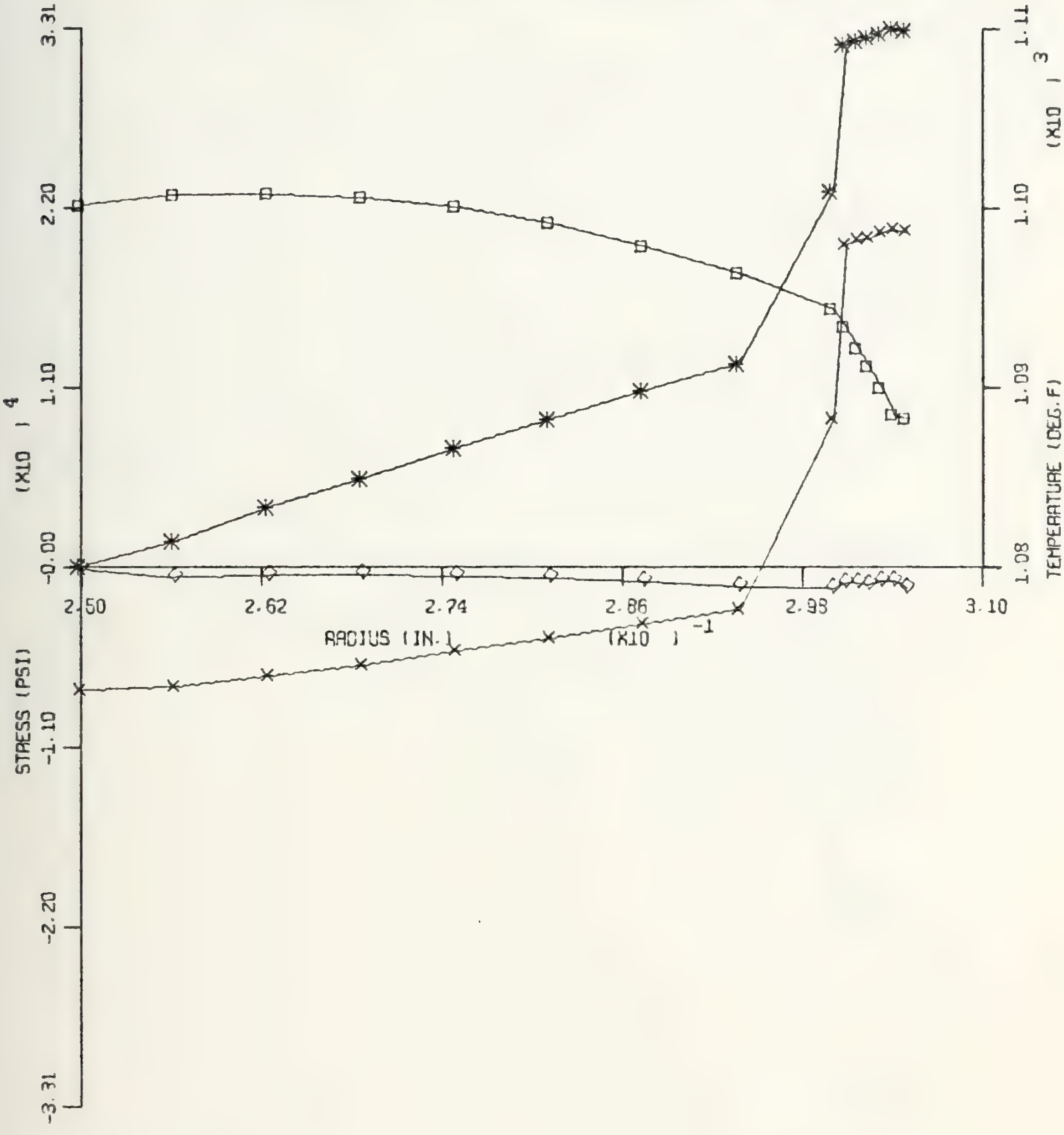
STRESS SCALE = 1.843X 10^{4} PSI/INCH

TEMPERATURE SCALE = 6.7 DEGREES/INCH

RADIAL POSITION SCALE = 0.01 INCHES/INCH

FIGURE 6(g)

STRESS AND TEMPERATURE
VERSUS RADIAL POSITION
TIME = 0.900 CONF.1-180



SIGMA R - DIAMOND SIGMA T - ASTERISK
SIGMA Z - CROSS TEMPERATURE - SQUARE

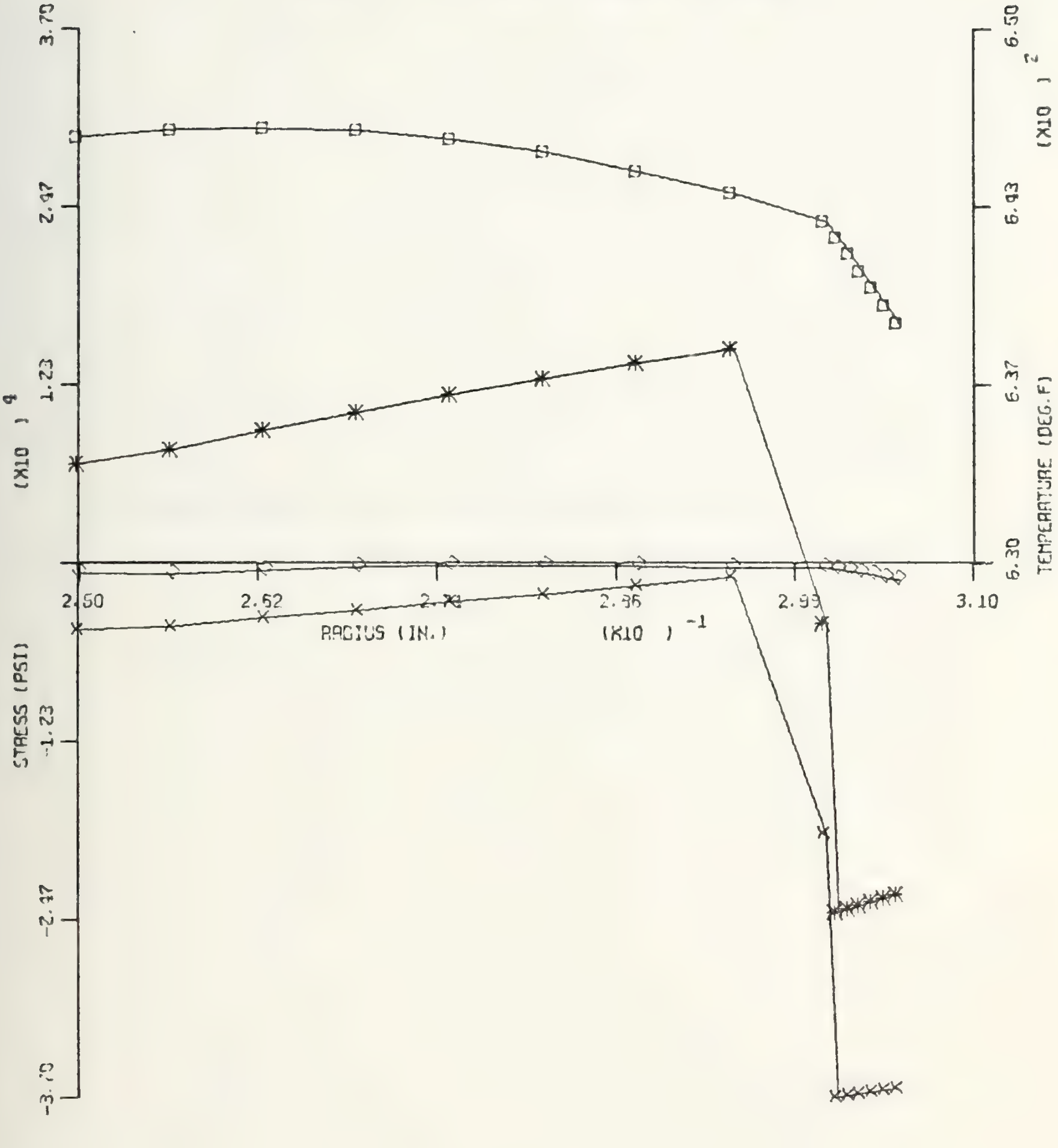
STRESS SCALE = 1.102X 10^4 PSI/INCH

TEMPERATURE SCALE = 10.0 DEGREES/INCH

RADIAL POSITION SCALE = 0.01 INCHES/INCH

FIGURE 6(h)

STRESS AND TEMPERATURE
VERSUS RADIAL POSITION
TIME = 1.050 CONF.1-180



SIGMA R - DIAMOND SIGMA T - ASTERISK
SIGMA Z - CROSS TEMPERATURE - SQUARE

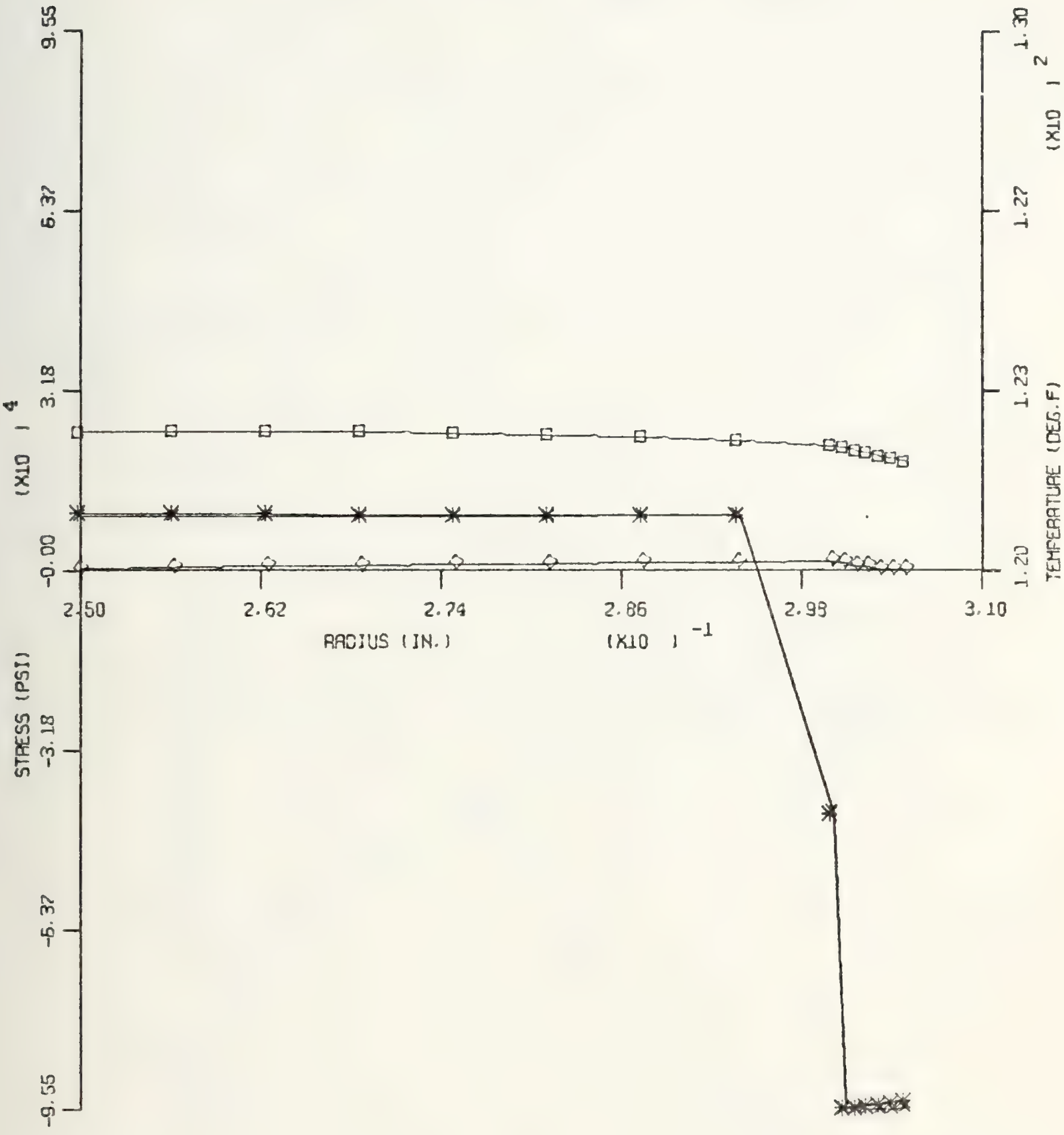
STRESS SCALE = 1.234X 10[4] PSI/INCH

TEMPERATURE SCALE = 6.7 DEGREES/INCH

RADIAL POSITION SCALE = 0.01 INCHES/INCH

FIGURE 6(i)

STRESS AND TEMPERATURE
VERSUS RADIAL POSITION
TIME = 1.650 CONF.1-180



SIGMA R - DIAMOND SIGMA T - ASTERISK
SIGMA Z - CROSS TEMPERATURE - SQUARE

STRESS SCALE = 3.184 X 10^{4} PSI/INCH

TEMPERATURE SCALE = 3.3 DEGREES/INCH

RADIAL POSITION SCALE = 0.01 INCHES/INCH

FIGURE 6(j)

<u>MAXIMUM AND MINIMUM PRINCIPAL STRESSES</u>
<u>VERSUS TIME</u>
<u>CONFIGURATION NUMBER ONE</u>

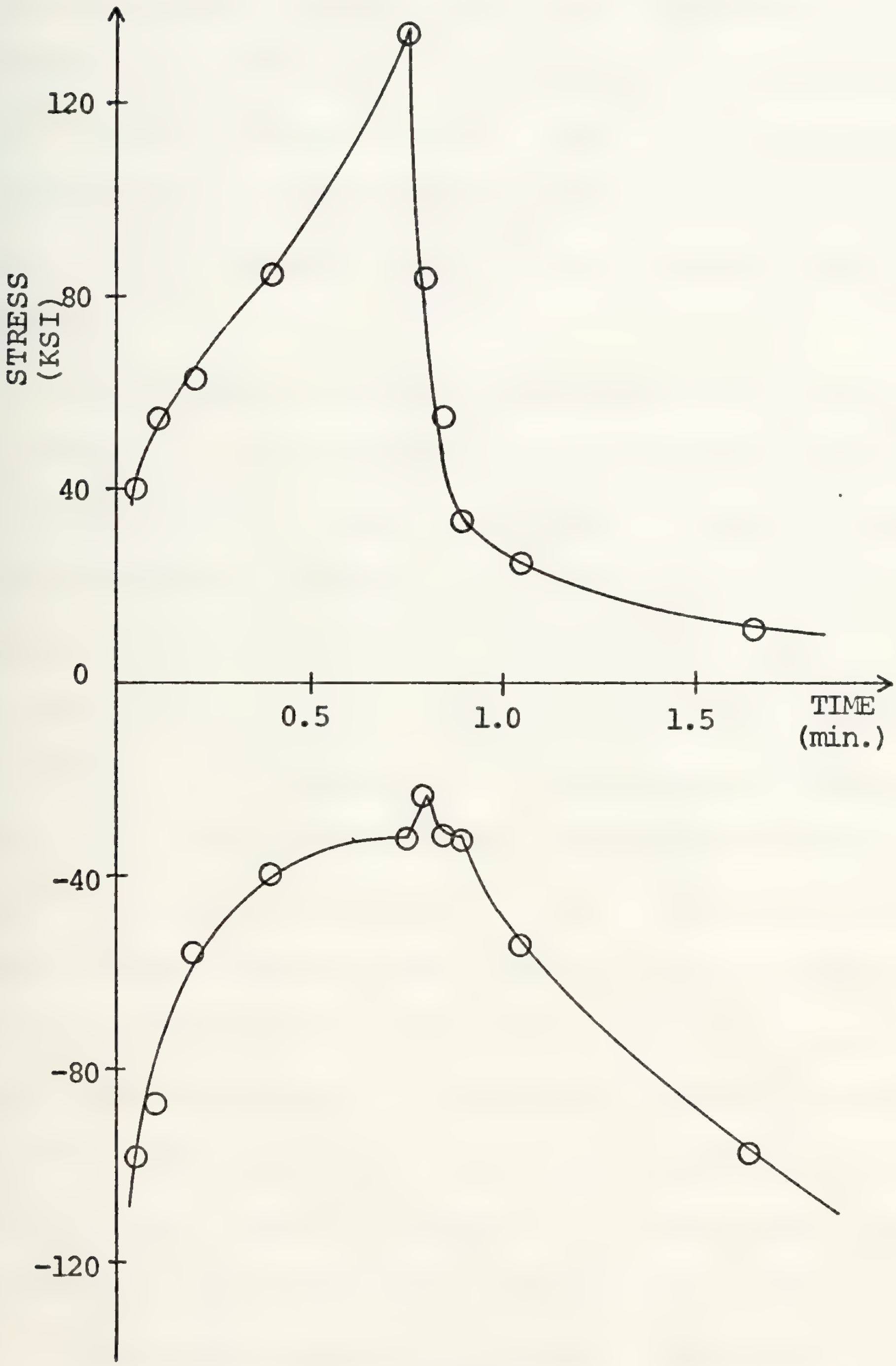


FIGURE 7

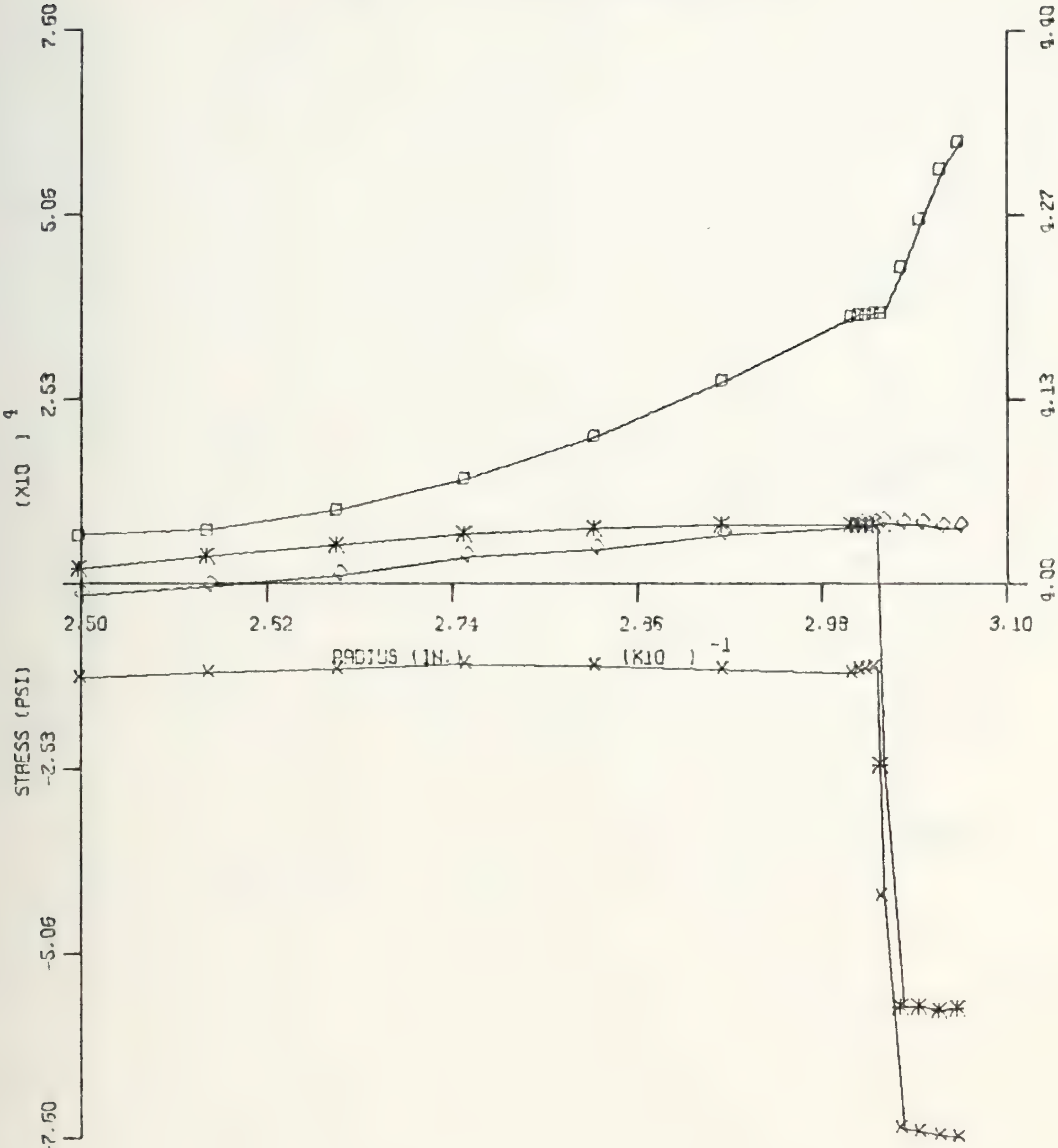
B. CONFIGURATION NUMBER TWO

This configuration contained a layer of pure nickel between the substratum and the layer of zirconia. The maximum principal stress (134.9 KSI) occurred at the maximum temperature condition and was located at a radius of 0.3045 inches, the zero angle position, and at the midpoint of the cylinder length. As with the other configurations, the larger tensile stresses were in the tangential and axial directions.

The comparison of stress and temperature distribution as a function of angle position is shown in Figure 12. The axial stress level is nearly constant as the circumferential position changes, however, the radial and circumferential stresses change at about 90 - 120 degrees. This change in the stress level coincides with a change in the temperature distribution. The change in the temperature distribution results from the changes in the temperature boundary conditions in the down stream portion of the air flow past the cylinder. Figure 12 was prepared using data for the cylinder ends at a radius of 0.3045 inches and a time of 0.75 minutes.

The time dependence of the stress levels is shown in Figure 13. The tangential and axial stresses for a position on the cylinder end at a radius of 0.3045 inches and an angle of 60° vary in the same manner as the temperature at that point. The radial stress is minimal throughout the time period.

STRESS AND TEMPERATURE
VERSUS RADIAL POSITION
TIME = 0.050 CONF.2-0



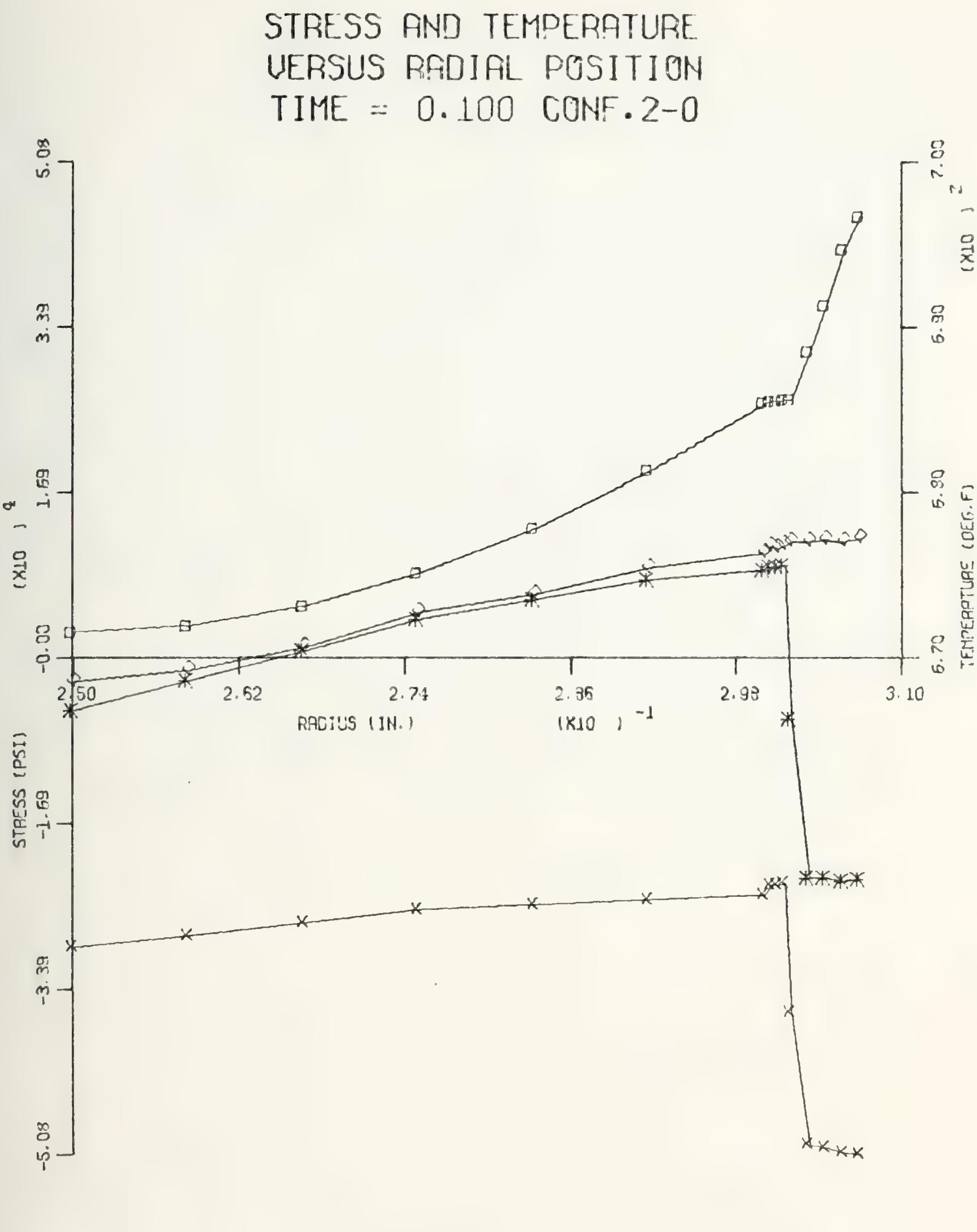
SIGMA R - DIAMOND SIGMA T - ASTERISK
SIGMA Z - CROSS TEMPERATURE - SQUARE

STRESS SCALE = 2.532K 10^4 PSI/INCH

TEMPERATURE SCALE = 13.3 DEGREES/INCH

RADIAL POSITION SCALE = 0.01 INCHES/INCH

FIGURE 8(a)



SIGMA R - DIAMOND SIGMA T - ASTERISK
SIGMA Z - CROSS TEMPERATURE - SQUARE

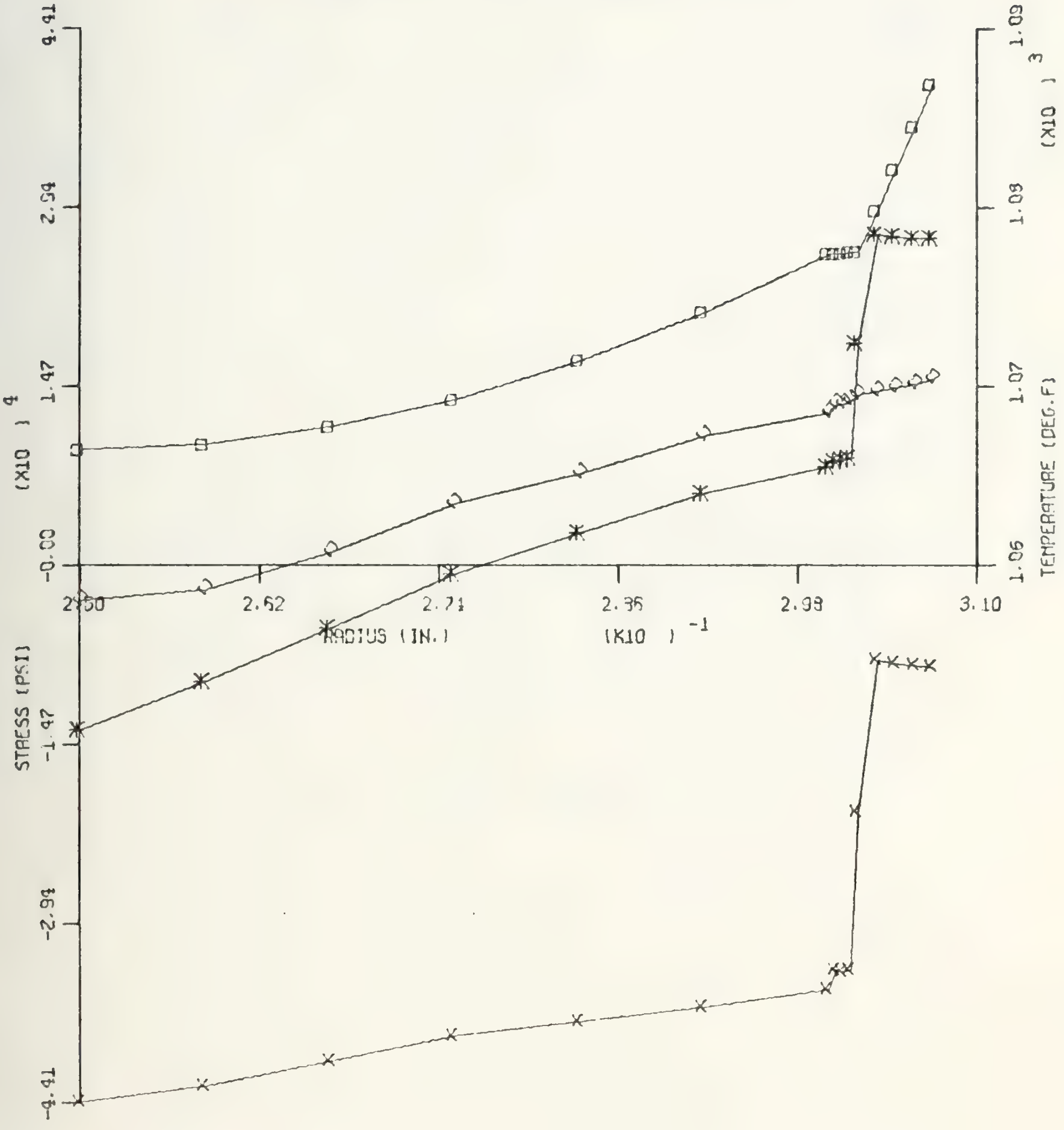
STRESS SCALE = 1.693X 10^4 PSI/INCH

TEMPERATURE SCALE = 10.0 DEGREES/INCH

RADIAL POSITION SCALE = 0.01 INCHES/INCH

FIGURE 8(b)

STRESS AND TEMPERATURE
VERSUS RADIAL POSITION
TIME = 0.200 CONF.2-0



SIGMA R - DIAMOND SIGMA T - ASTERISK
SIGMA Z - CROSS TEMPERATURE - SQUARE

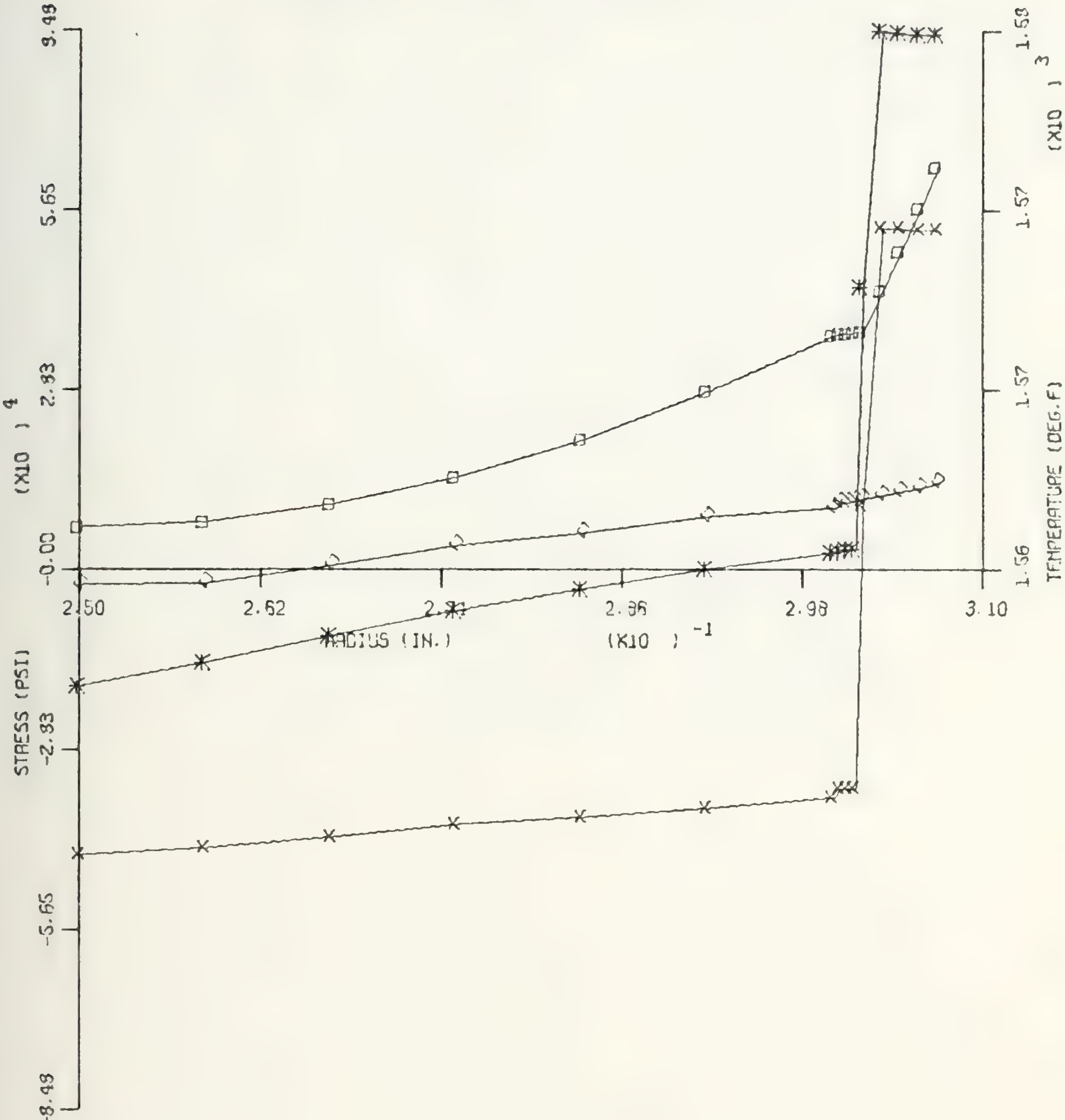
STRESS SCALE = 1.458X 10^4 PSI/INCH

TEMPERATURE SCALE = 10.0 DEGREES/INCH

RADIAL POSITION SCALE = 0.01 INCHES/INCH

FIGURE 8(c)

STRESS AND TEMPERATURE
VERSUS RADIAL POSITION
TIME = 0.400 CONF.2-0



SIGMA R - DIAMOND SIGMA T - ASTERISK
SIGMA Z - CROSS TEMPERATURE - SQUARE

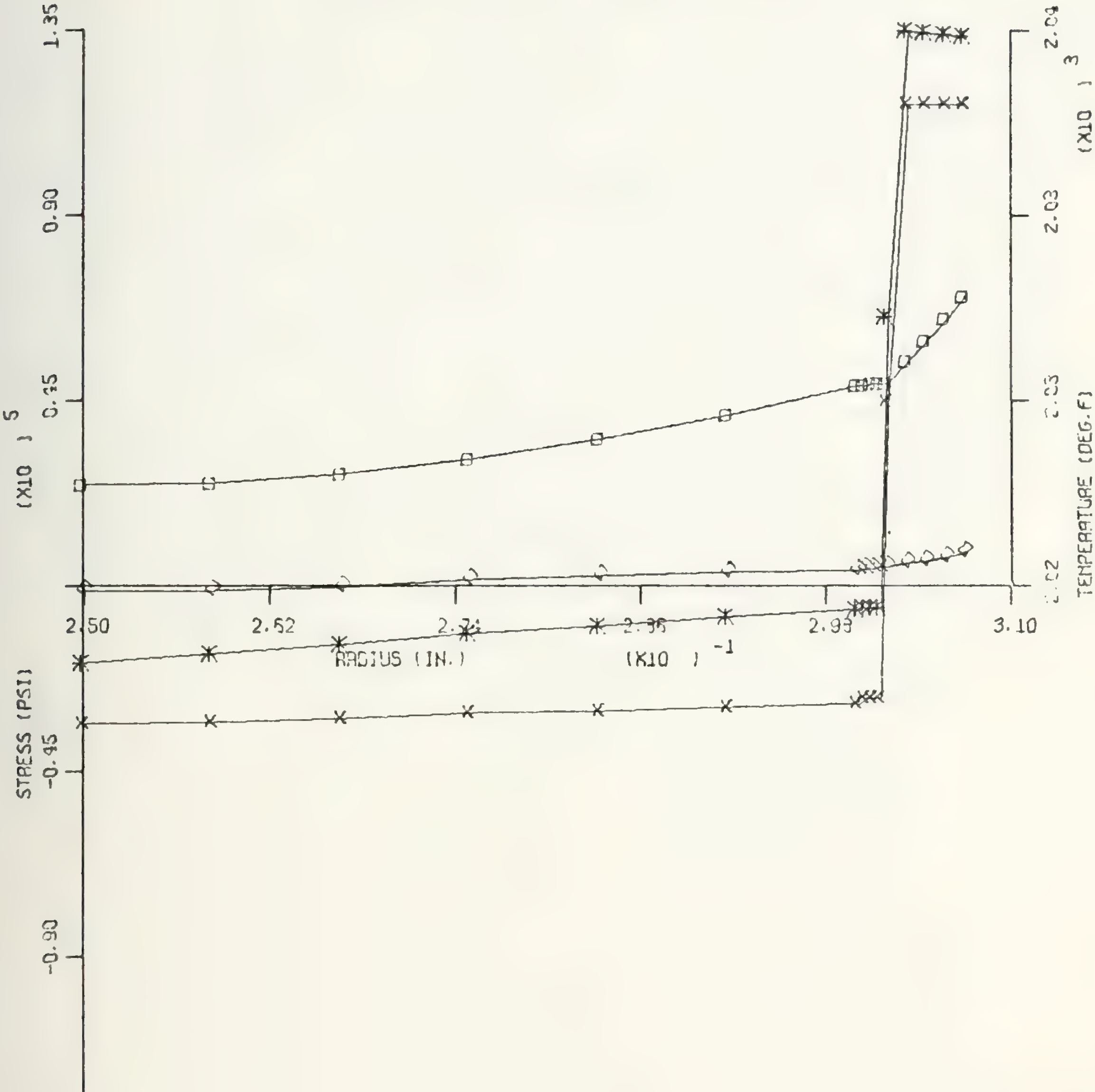
STRESS SCALE = 2.825X 10^4 PSI/INCH

TEMPERATURE SCALE = 6.7 DEGREES/INCH

RADIAL POSITION SCALE = 0.01 INCHES/INCH

FIGURE 8(d)

STRESS AND TEMPERATURE
VERSUS RADIAL POSITION
TIME = 0.750 CONF.2-0



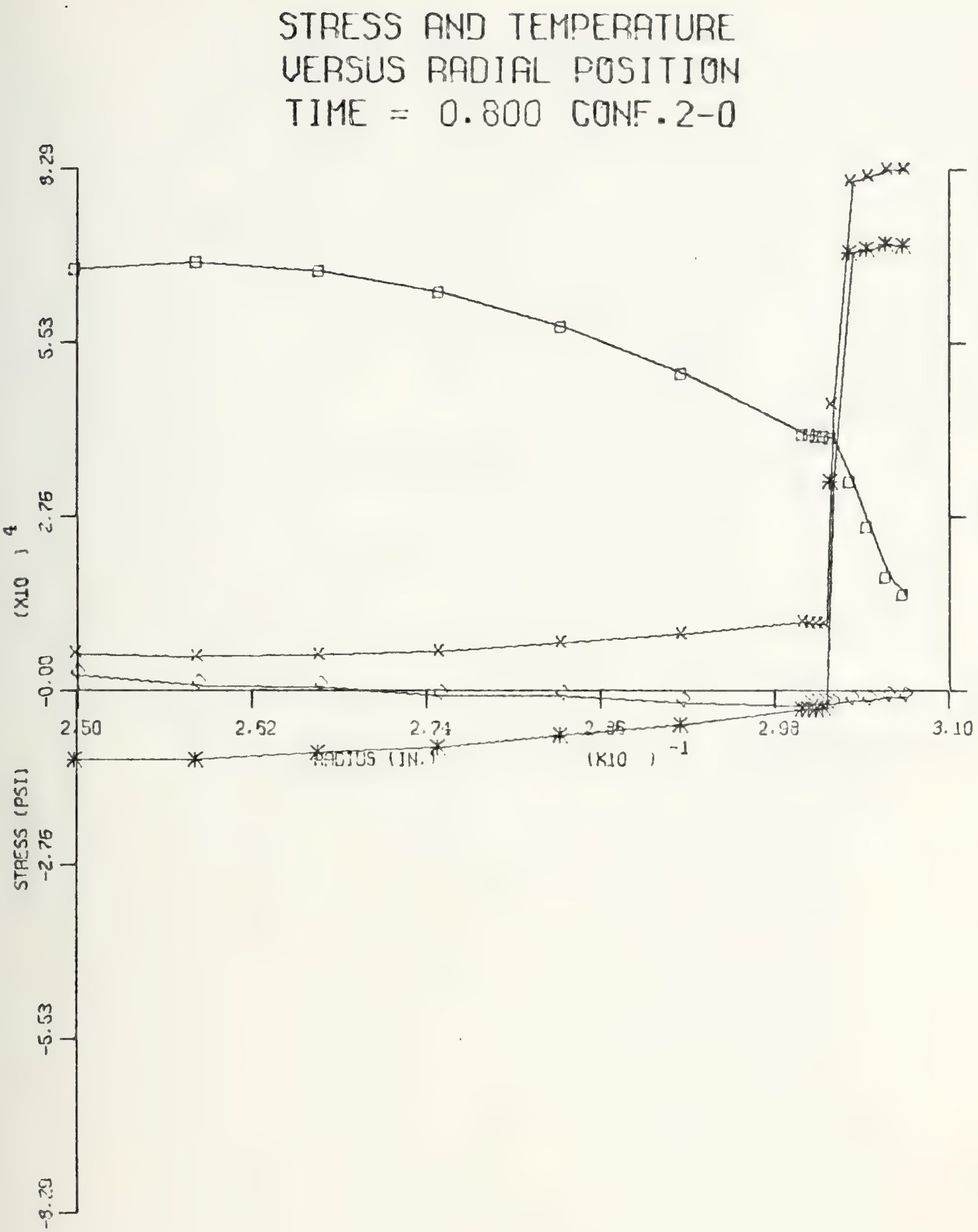
SIGMA R - DIAMOND SIGMA T - ASTERISK
SIGMA Z - CROSS TEMPERATURE - SQUARE

STRESS SCALE = 4.512X 10^4 PSI/INCH

TEMPERATURE SCALE = 6.7 DEGREES/INCH

RADIAL POSITION SCALE = 0.01 INCHES/INCH

FIGURE 8(e)



SIGMA R - DIAMOND SIGMA T - ASTERISK
SIGMA Z - CROSS TEMPERATURE - SQUARE

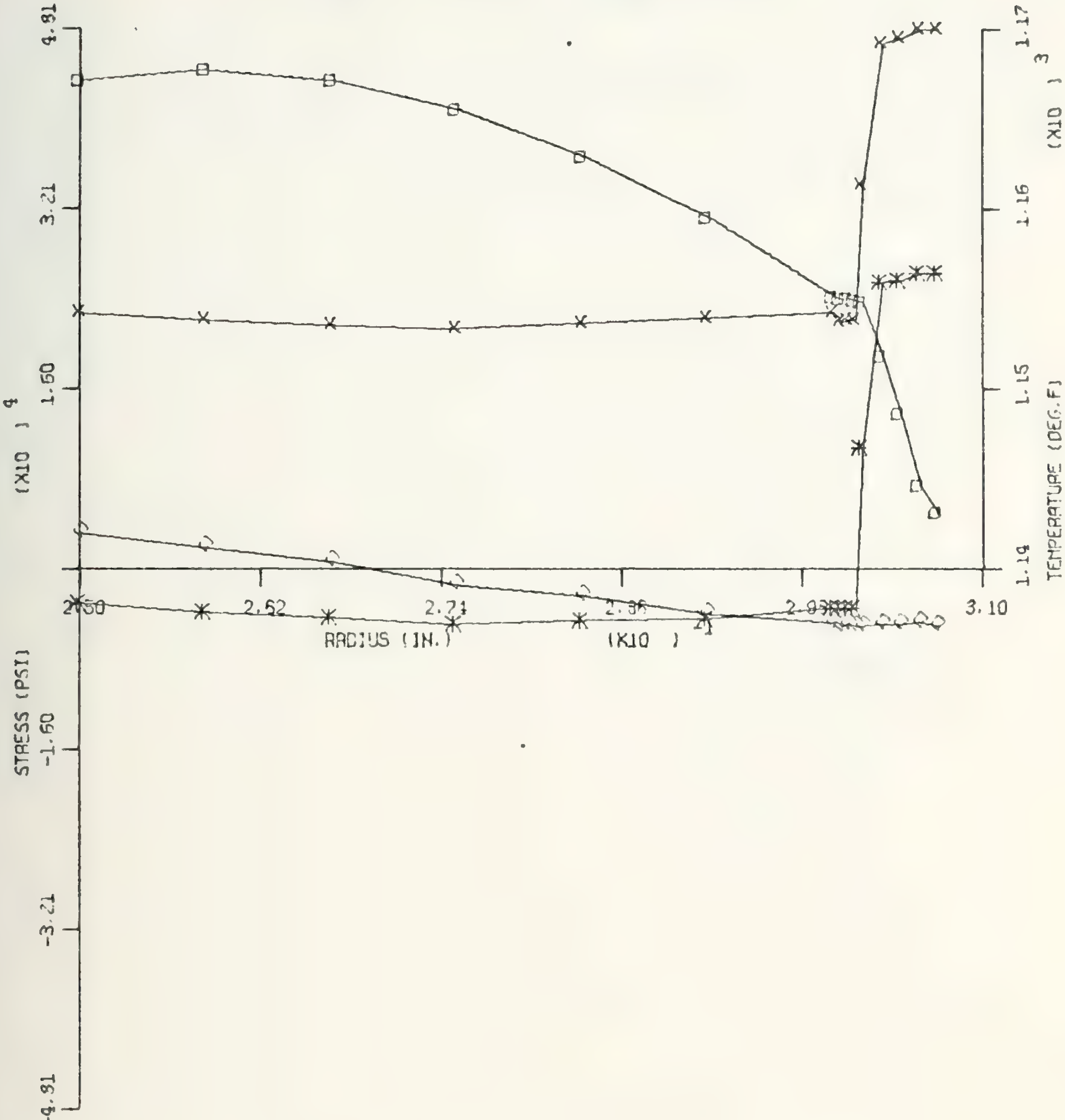
STRESS SCALE = 2.764×10^4 PSI/INCH

TEMPERATURE SCALE = 16.7 DEGREES/INCH

RADIAL POSITION SCALE = 0.01 INCHES/INCH

FIGURE 8(f)

STRESS AND TEMPERATURE
VERSUS RADIAL POSITION
TIME = 0.850 CONF.2-0



SIGMA R - DIAMOND SIGMA T - ASTERISK
SIGMA Z - CROSS TEMPERATURE - SQUARE

STRESS SCALE = 1.604X 10^4 PSI/INCH

TEMPERATURE SCALE = 10.0 DEGREES/INCH

RADIAL POSITION SCALE = 0.01 INCHES/INCH

FIGURE 8(g)

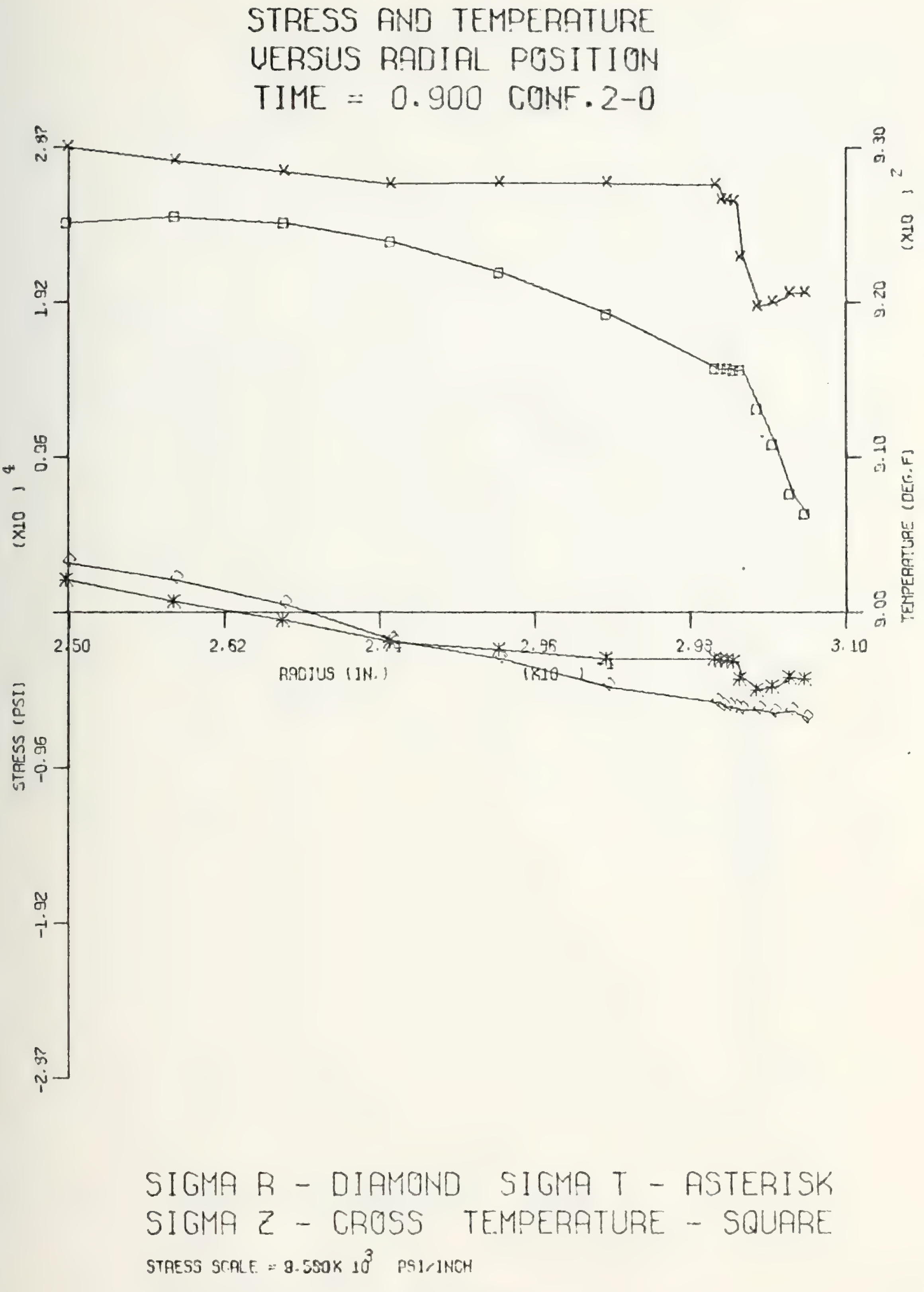
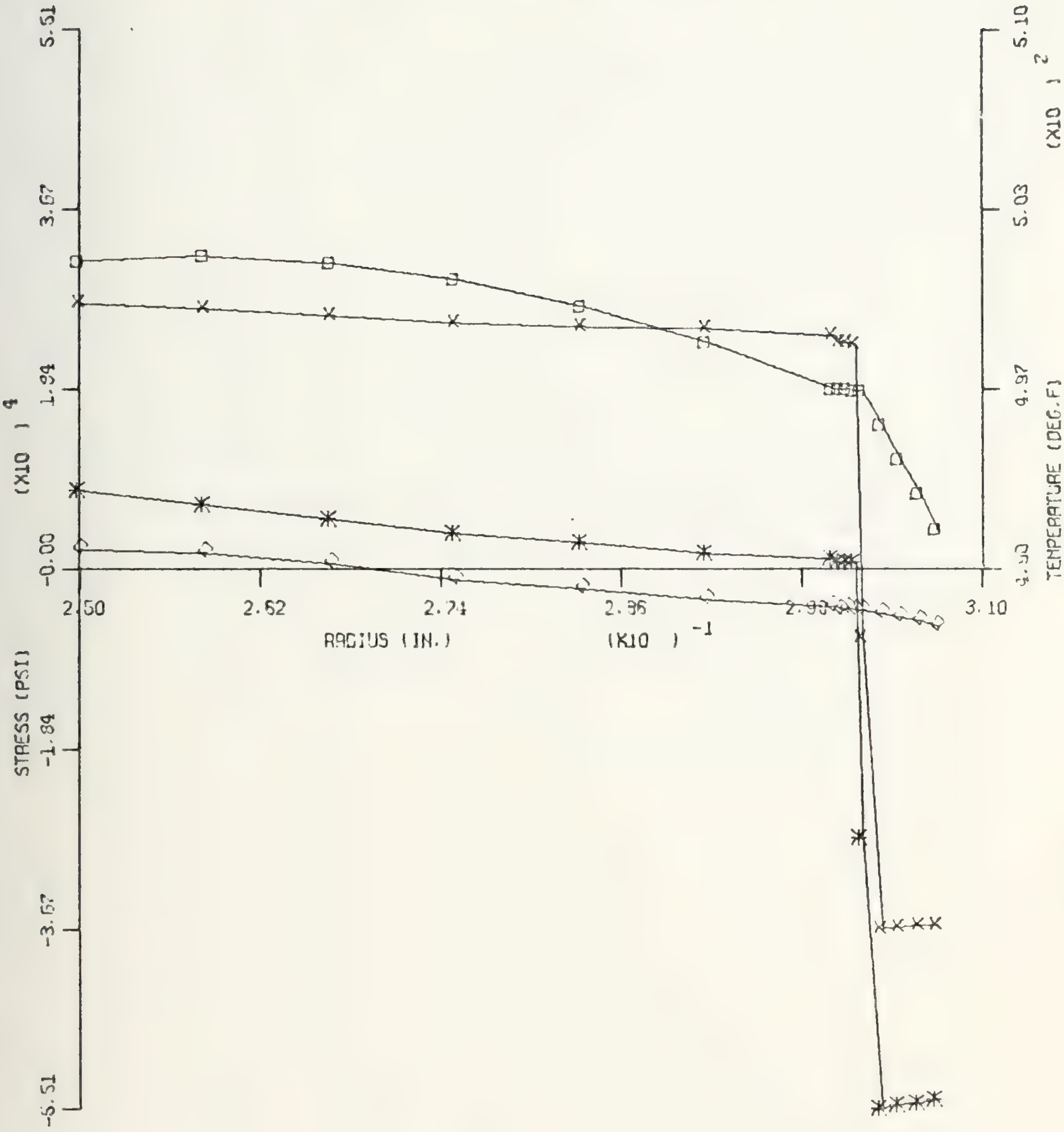

STRESS AND TEMPERATURE
VERSUS RADIAL POSITION
TIME = 0.900 CONF.2-0
RADIUS (IN.)
STRESS (PSI)
TEMPERATURE (DEG.F)
SIGMA R - DIAMOND SIGMA T - ASTERISK
SIGMA Z - CROSS TEMPERATURE - SQUARE
STRESS SCALE = 8.550X 10^3 PSI/INCH
TEMPERATURE SCALE = 10.0 DEGREES/INCH
RADIAL POSITION SCALE = 0.01 INCHES/INCH

FIGURE 8(h)

STRESS AND TEMPERATURE
VERSUS RADIAL POSITION
TIME = 1.050 CONF.2-0



SIGMA R - DIAMOND SIGMA T - ASTERISK
SIGMA Z - CROSS TEMPERATURE - SQUARE

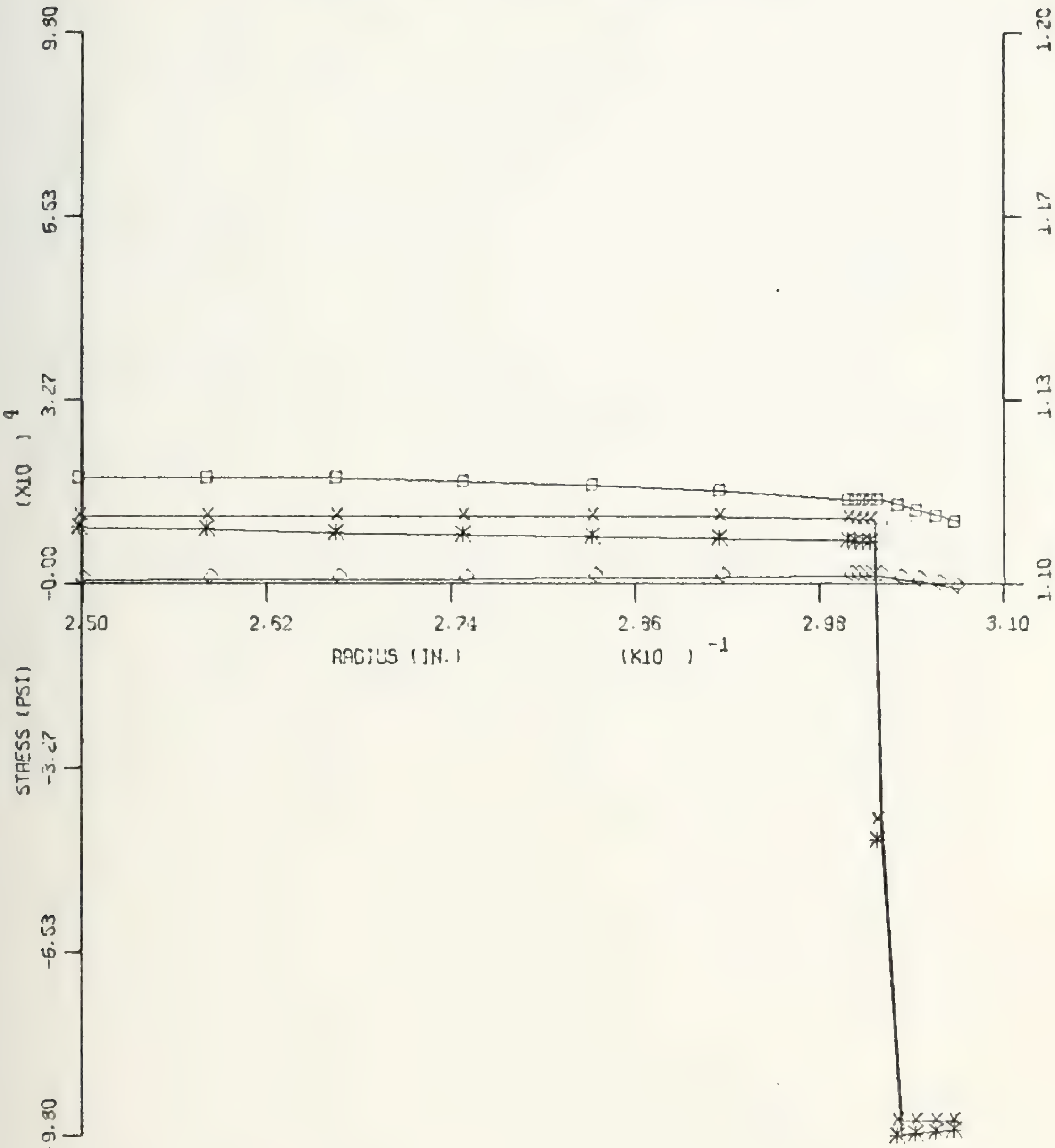
STRESS SCALE = 1.835X 10^4 PSI/INCH

TEMPERATURE SCALE = 6.7 DEGREES/INCH

RADIAL POSITION SCALE = 0.01 INCHES/INCH

FIGURE 8(i)

STRESS AND TEMPERATURE
VERSUS RADIAL POSITION
TIME = 1.650 CONF.2-0



SIGMA R - DIAMOND SIGMA T - ASTERISK
SIGMA Z - CROSS TEMPERATURE - SQUARE

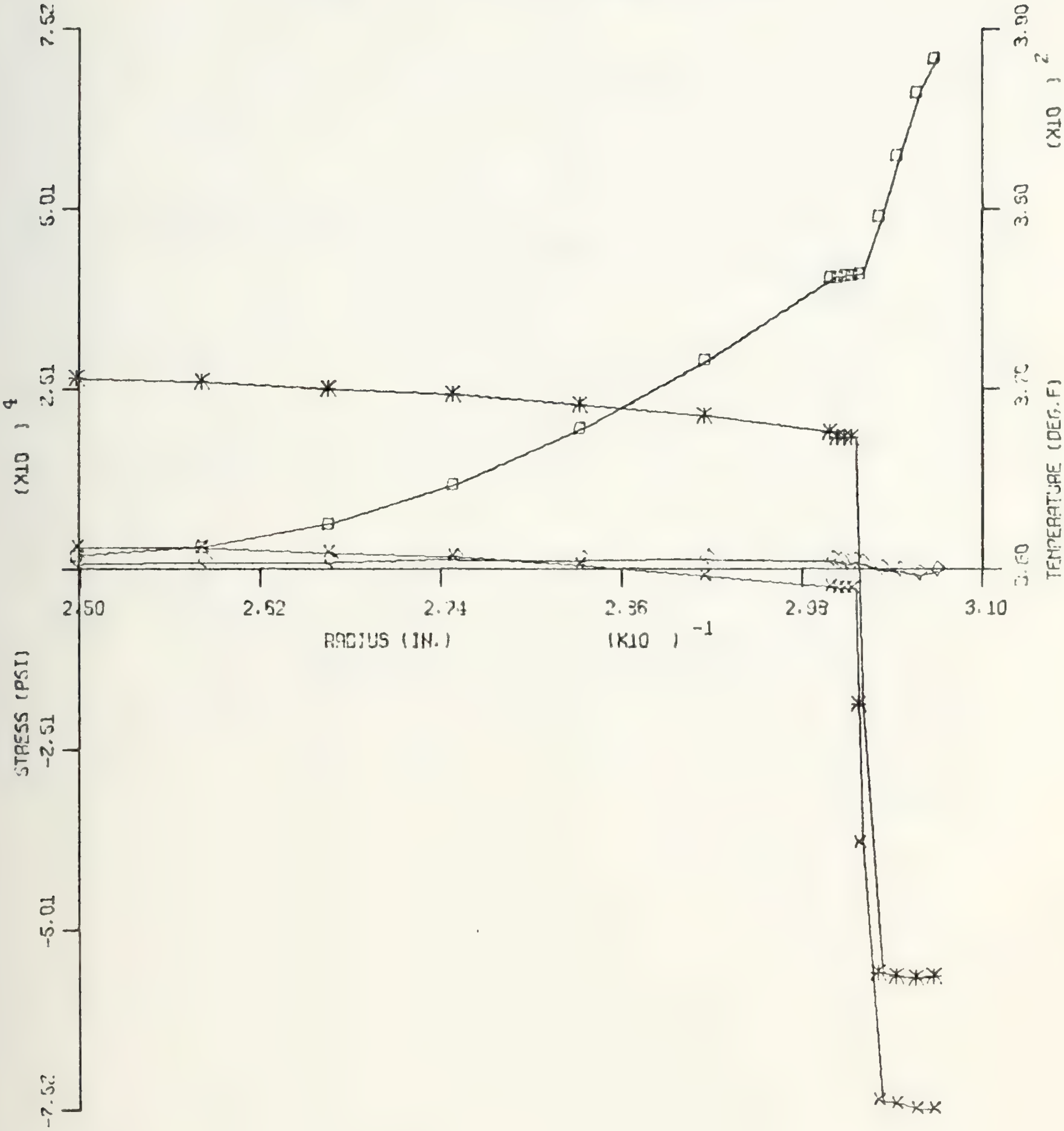
STRESS SCALE = 3.265X 10^4 PSI/INCH

TEMPERATURE SCALE = 3.3 DEGREES/INCH

RADIAL POSITION SCALE = 0.01 INCHES/INCH

FIGURE 8(j)

STRESS AND TEMPERATURE
VERSUS RADIAL POSITION
TIME = 0.050 CONF.2-60



SIGMA R - DIAMOND SIGMA T - ASTERISK
SIGMA Z - CROSS TEMPERATURE - SQUARE

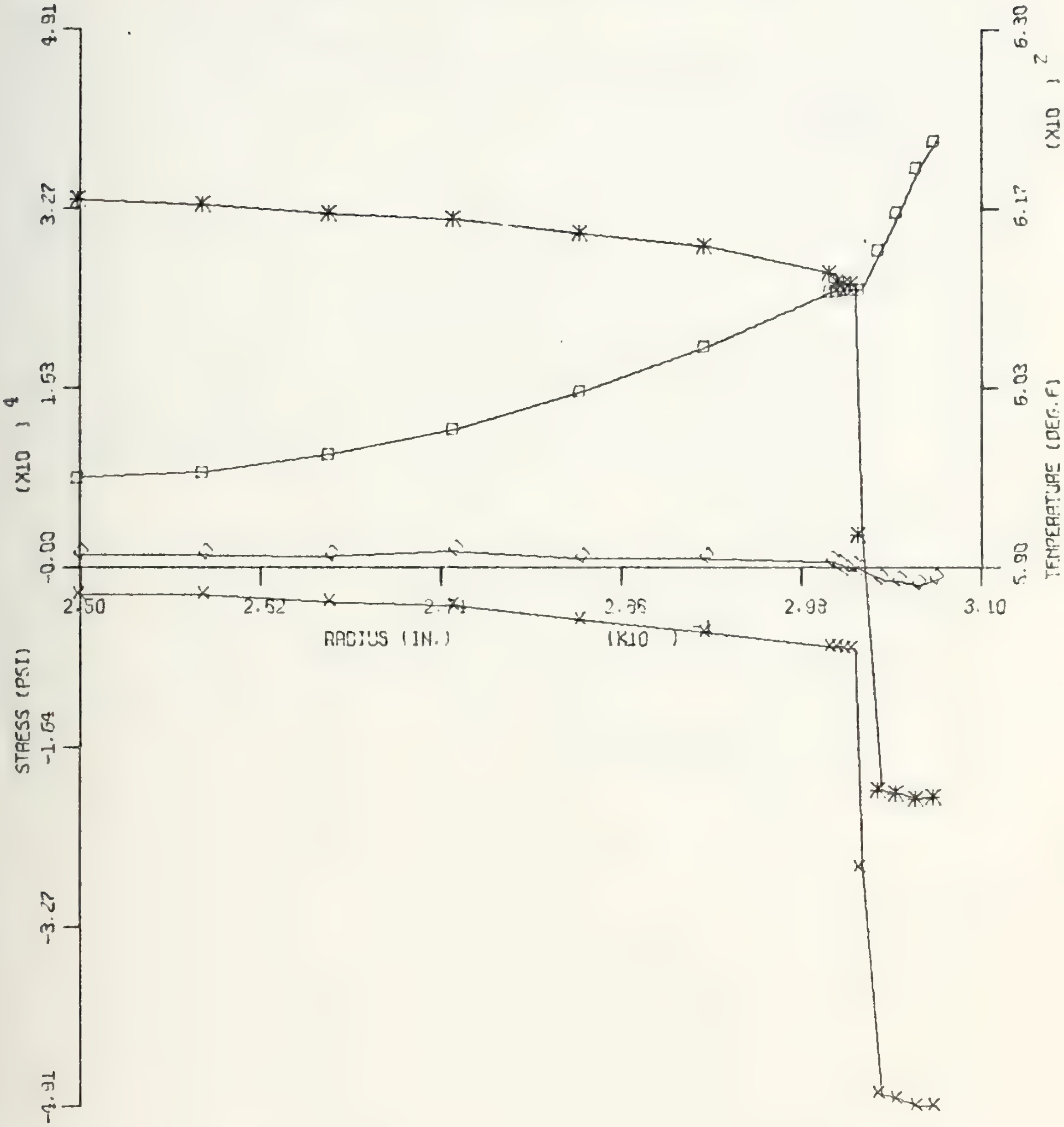
STRESS SCALE = 2.507X 10³ PSI/INCH

TEMPERATURE SCALE = 10.0 DEGREES/INCH

RADIAL POSITION SCALE = 0.01 INCHES/INCH

FIGURE 9(a)

STRESS AND TEMPERATURE
VERSUS RADIAL POSITION
TIME = 0.100 CONF.2-60



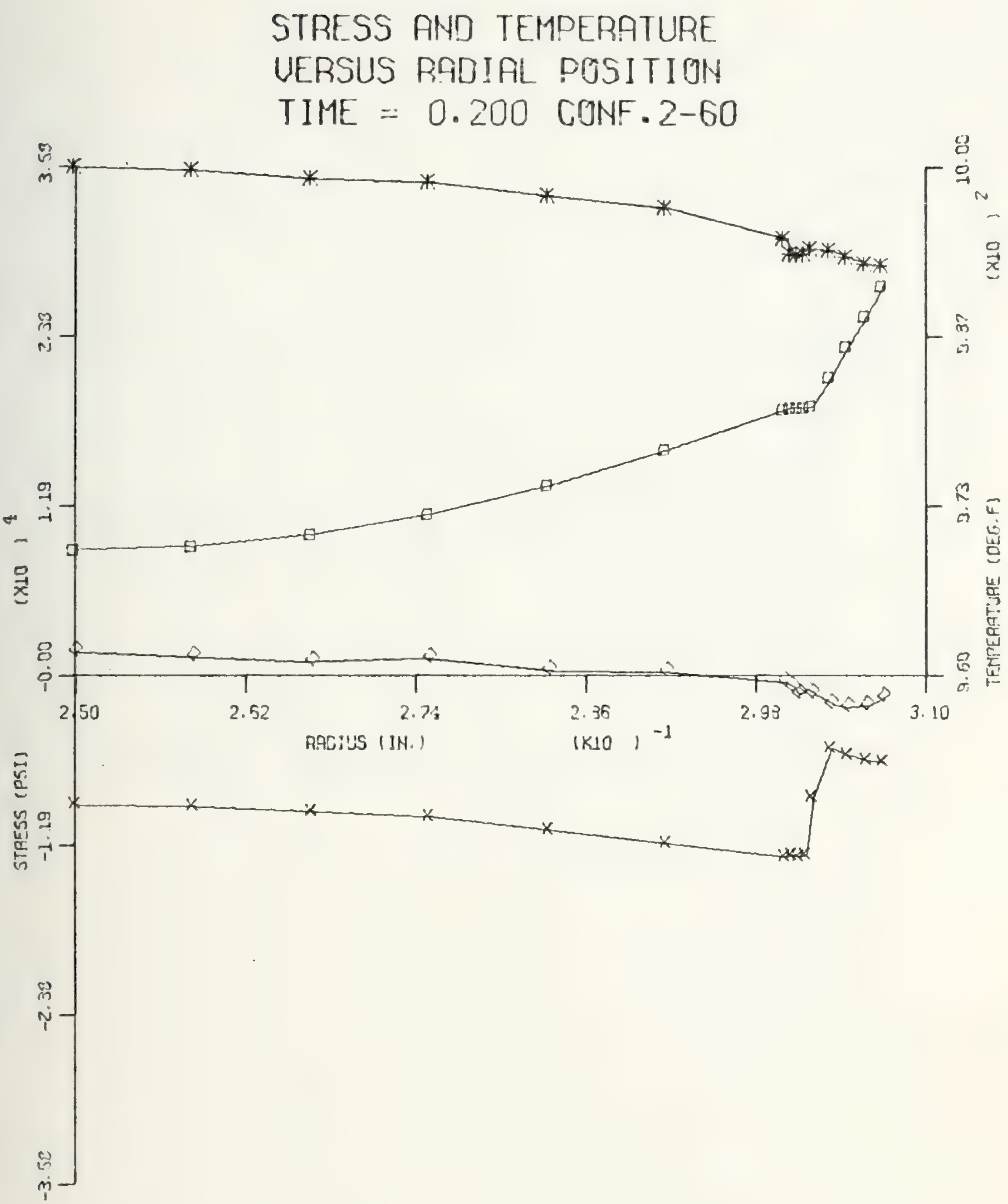
SIGMA R - DIAMOND SIGMA T - ASTERISK
SIGMA Z - CROSS TEMPERATURE - SQUARE

STRESS SCALE = 1.635×10^4 PSI/INCH

TEMPERATURE SCALE = 13.3 DEGREES/INCH

RADIAL POSITION SCALE = 0.01 INCHES/INCH

FIGURE 9(b)



SIGMA R - DIAMOND SIGMA T - ASTERISK
SIGMA Z - CROSS TEMPERATURE - SQUARE

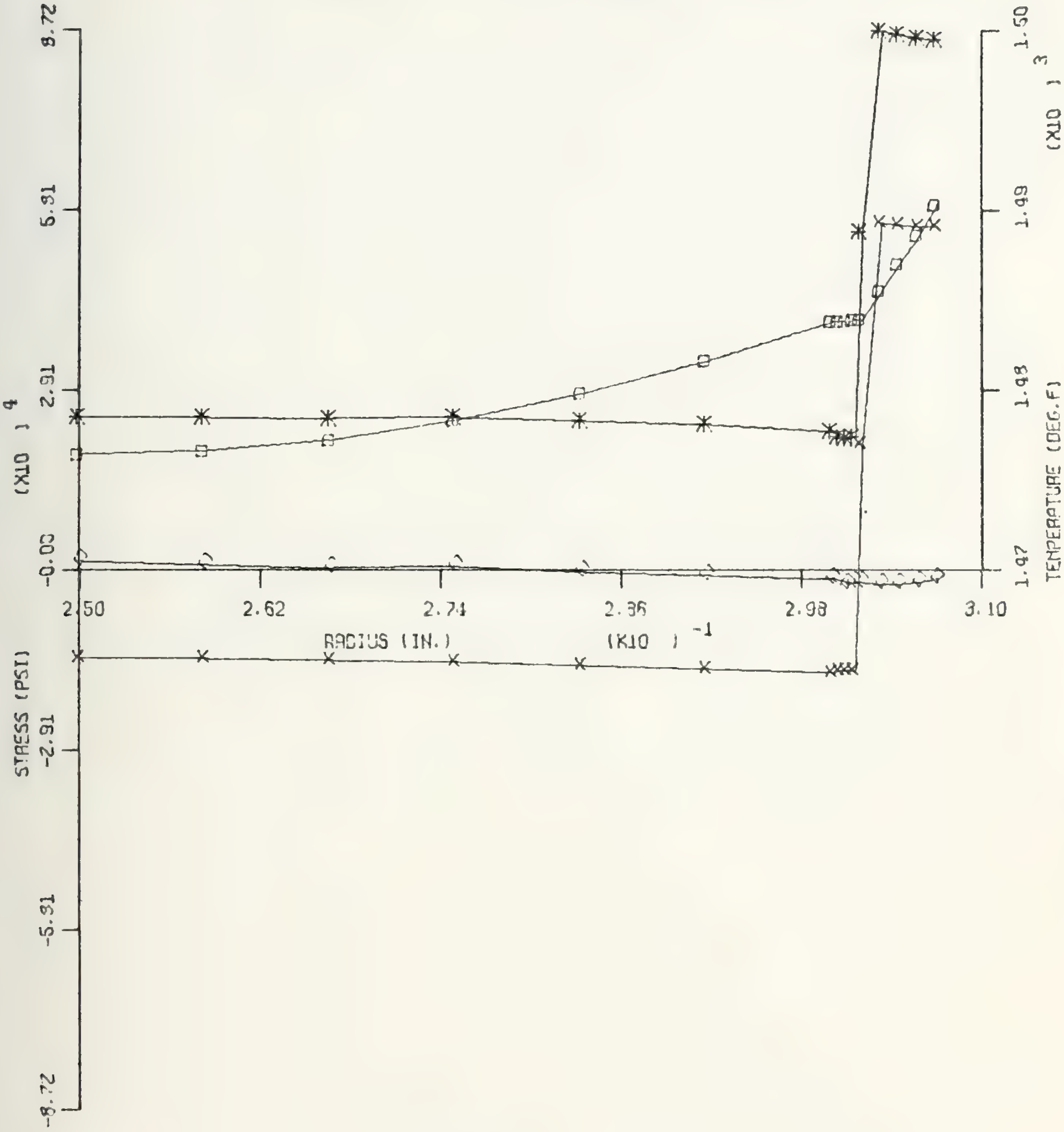
STRESS SCALE = 1.192X 10^{4} PSI/INCH

TEMPERATURE SCALE = 13.3 DEGREES/INCH

RADIAL POSITION SCALE = 0.01 INCHES/INCH

FIGURE 9(c)

STRESS AND TEMPERATURE
VERSUS RADIAL POSITION
TIME = 0.400 CONF.2-60



SIGMA R - DIAMOND SIGMA T - ASTERISK
SIGMA Z - CROSS TEMPERATURE - SQUARE

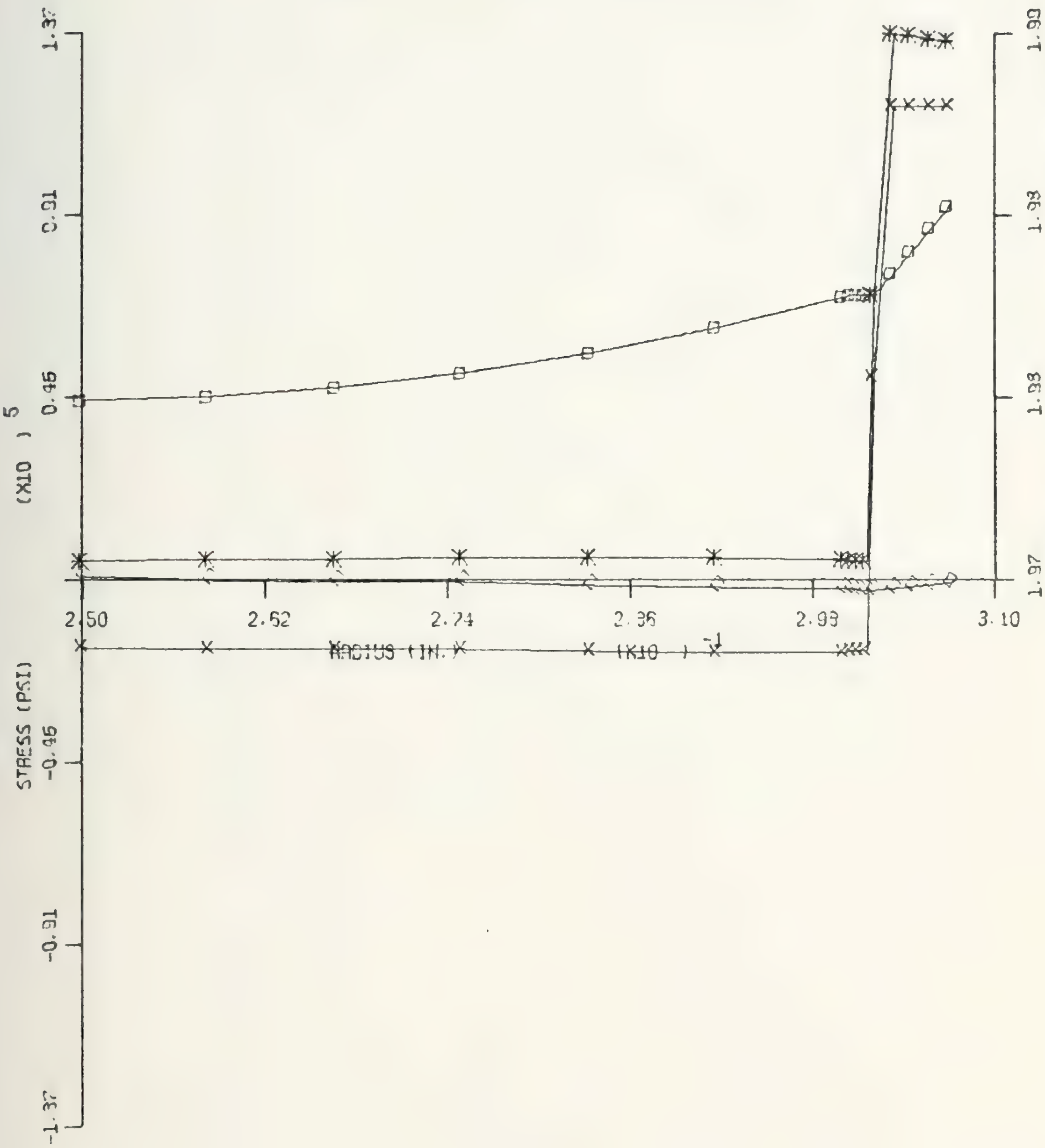
STRESS SCALE = 2.905K 10^4 PSI/INCH

TEMPERATURE SCALE = 10.0 DEGREES/INCH

RADIAL POSITION SCALE = 0.01 INCHES/INCH

FIGURE 9(d)

STRESS AND TEMPERATURE
VERSUS RADIAL POSITION
TIME = 0.750 CONF.2-60



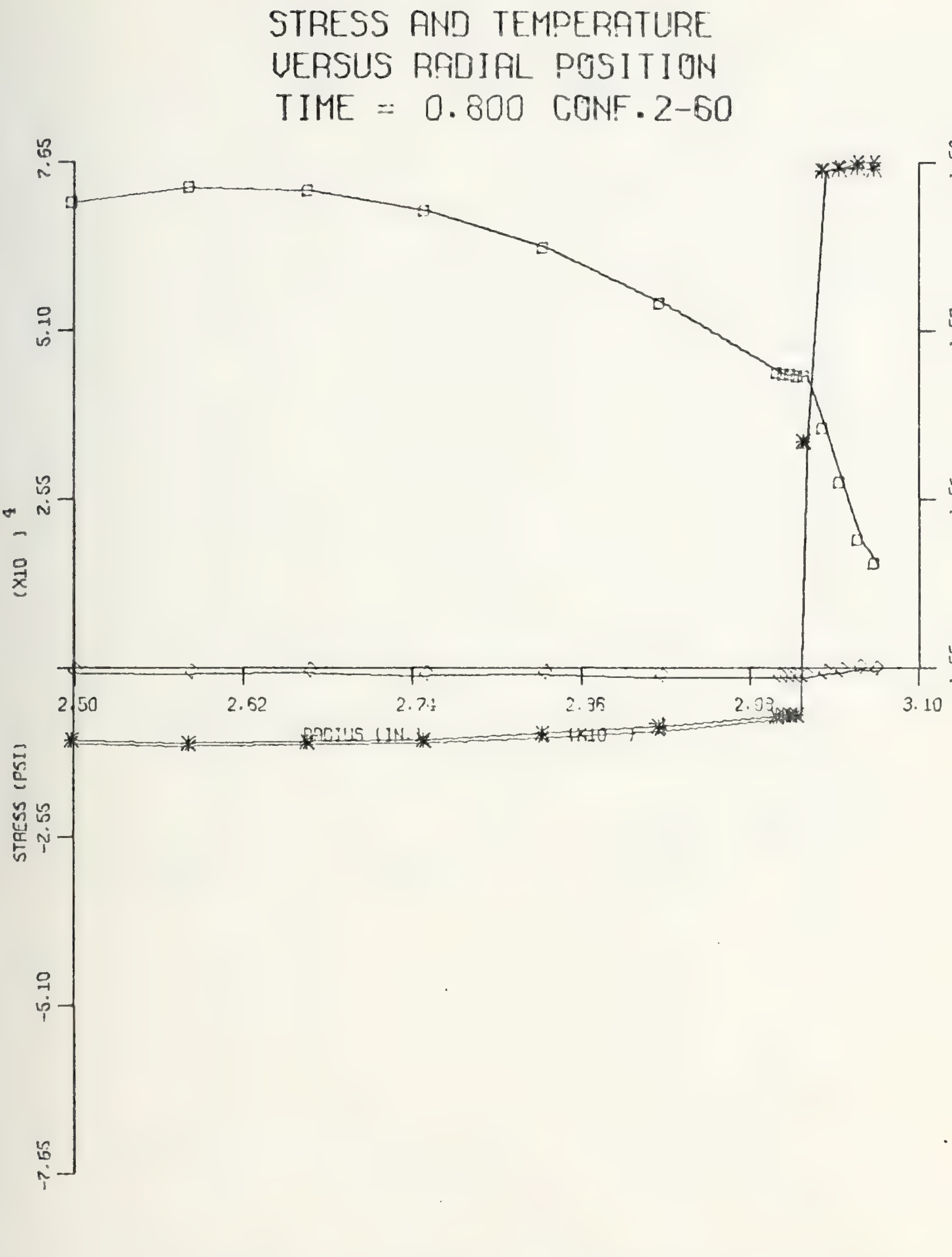
SIGMA R - DIAMOND SIGMA T - ASTERISK
SIGMA Z - CROSS TEMPERATURE - SQUARE

STRESS SCALE = 4.556K 10^4 PSI/INCH

TEMPERATURE SCALE = 6.7 DEGREES/INCH

RADIAL POSITION SCALE = 0.01 INCHES/INCH

FIGURE 9(e)



SIGMA R - DIAMOND SIGMA T - ASTERISK
SIGMA Z - CROSS TEMPERATURE - SQUARE

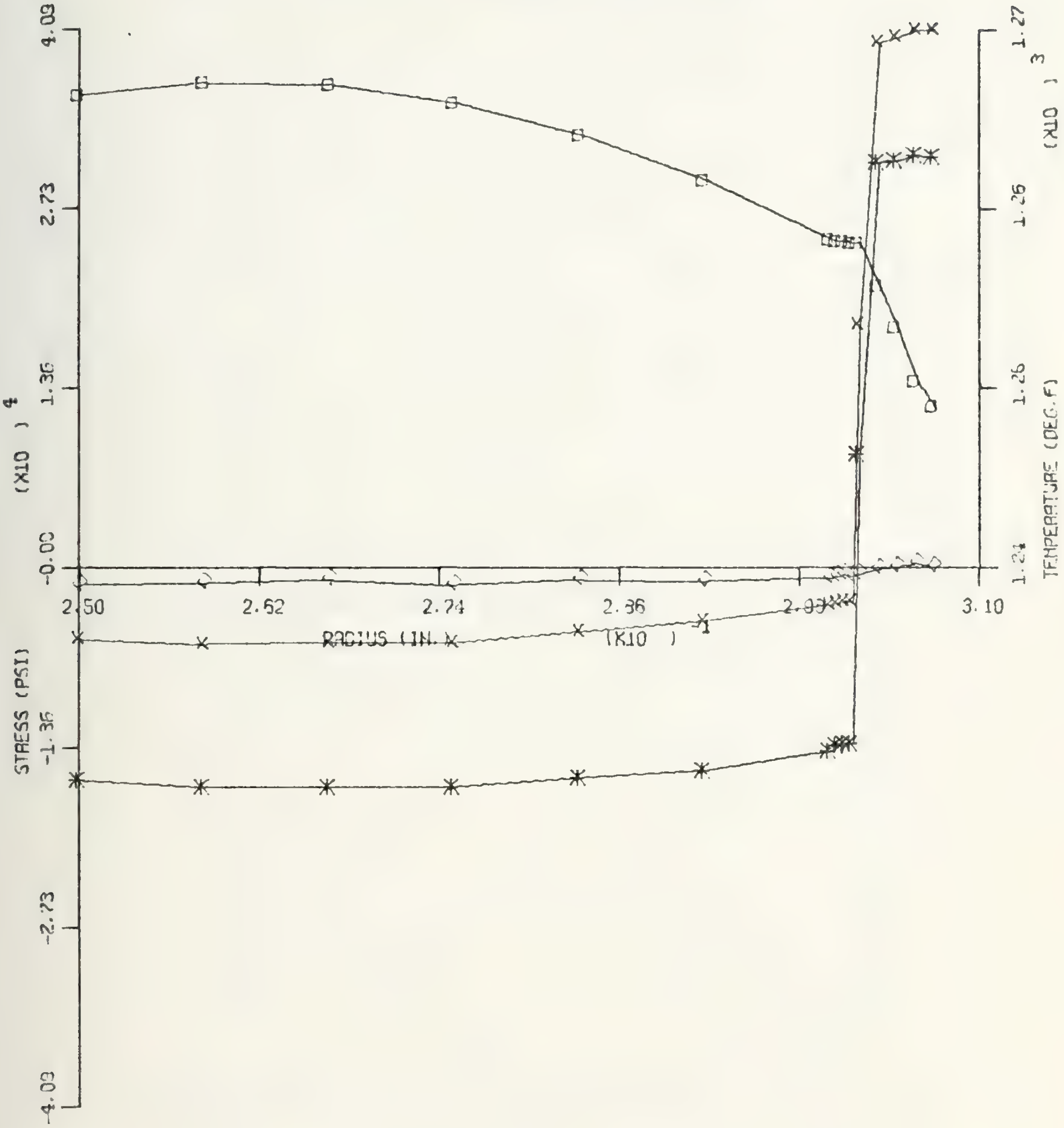
STRESS SCALE = 2.550×10^{4} PSI/INCH

TEMPERATURE SCALE = 10.0 DEGREES/INCH

RADIAL POSITION SCALE = 0.01 INCHES/INCH

FIGURE 9(f)

STRESS AND TEMPERATURE
VERSUS RADIAL POSITION
TIME = 0.850 CONF.2-60



SIGMA R - DIAMOND SIGMA T - ASTERISK
SIGMA Z - CROSS TEMPERATURE - SQUARE

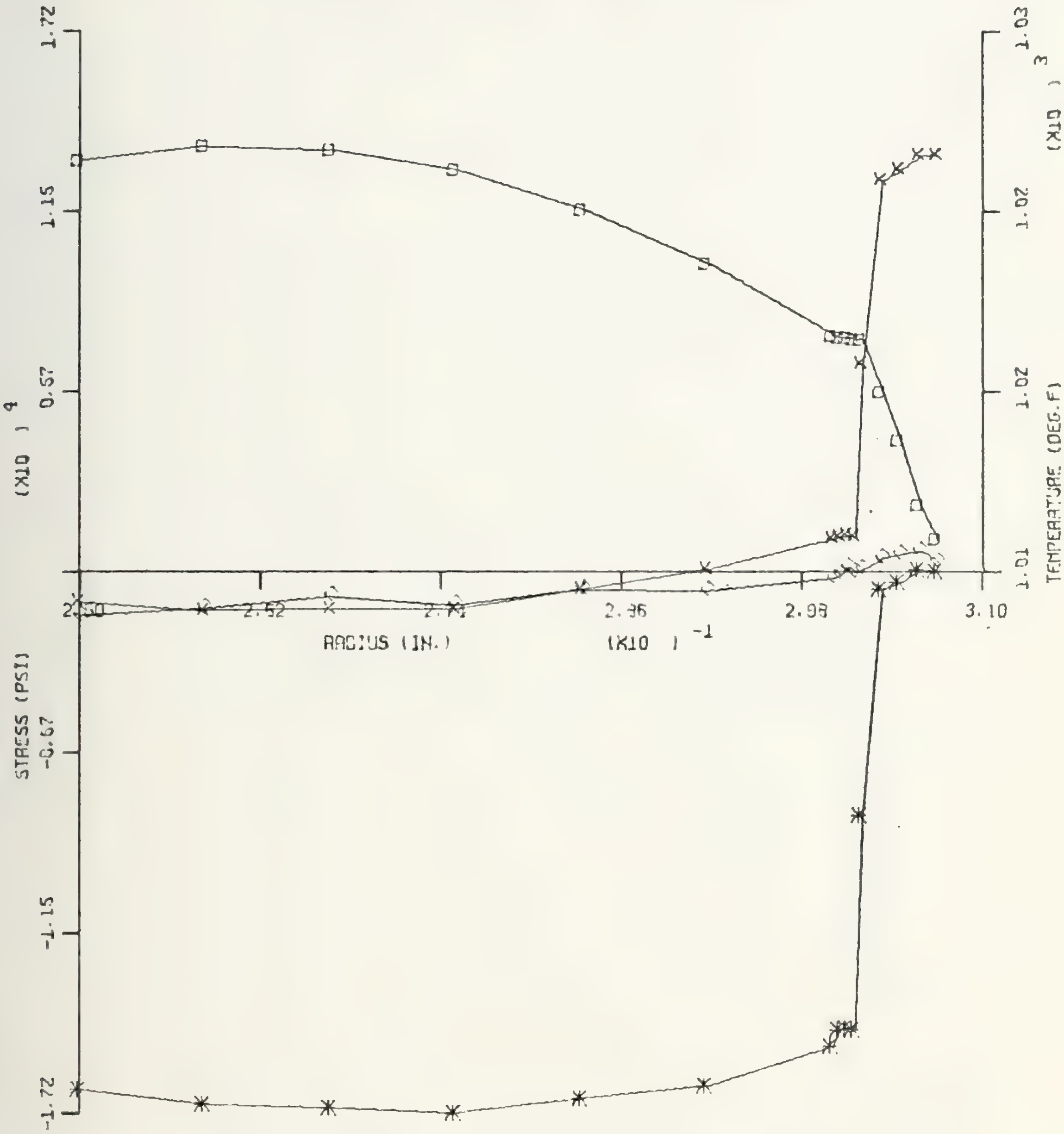
STRESS SCALE = 1.364 X 10^{3} PSI/INCH

TEMPERATURE SCALE = 10.0 DEGREES/INCH

RADIAL POSITION SCALE = 0.01 INCHES/INCH

FIGURE 9(g)

STRESS AND TEMPERATURE
VERSUS RADIAL POSITION
TIME = 0.900 CONF.2-60



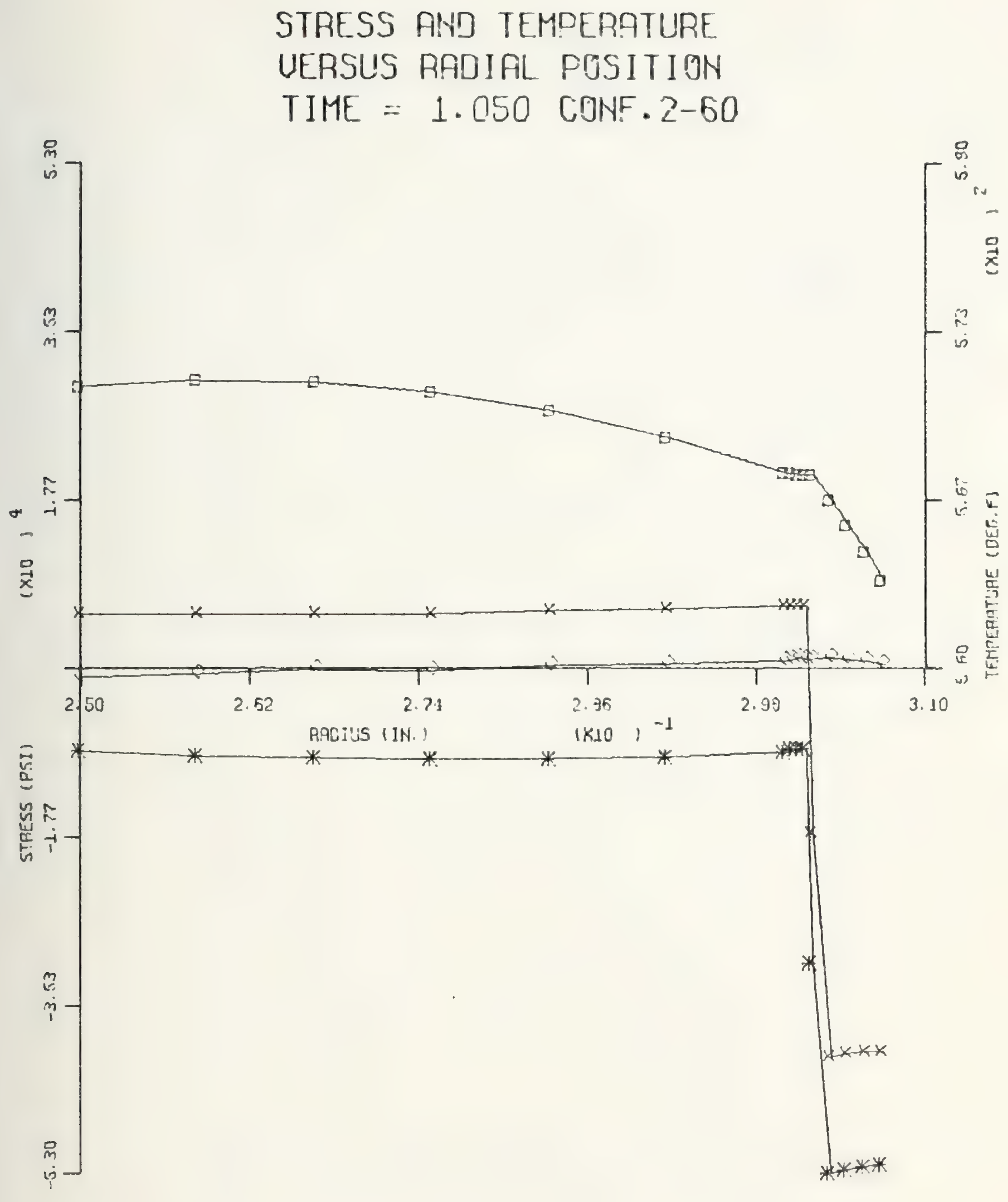
SIGMA R - DIAMOND SIGMA T - ASTERISK
SIGMA Z - CROSS TEMPERATURE - SQUARE

STRESS SCALE = 5.723K 10^3 PSI/INCH

TEMPERATURE SCALE = 6.7 DEGREES/INCH

RADIAL POSITION SCALE = 0.01 INCHES/INCH

FIGURE 9(h)



SIGMA R - DIAMOND SIGMA T - ASTERISK
SIGMA Z - CROSS TEMPERATURE - SQUARE

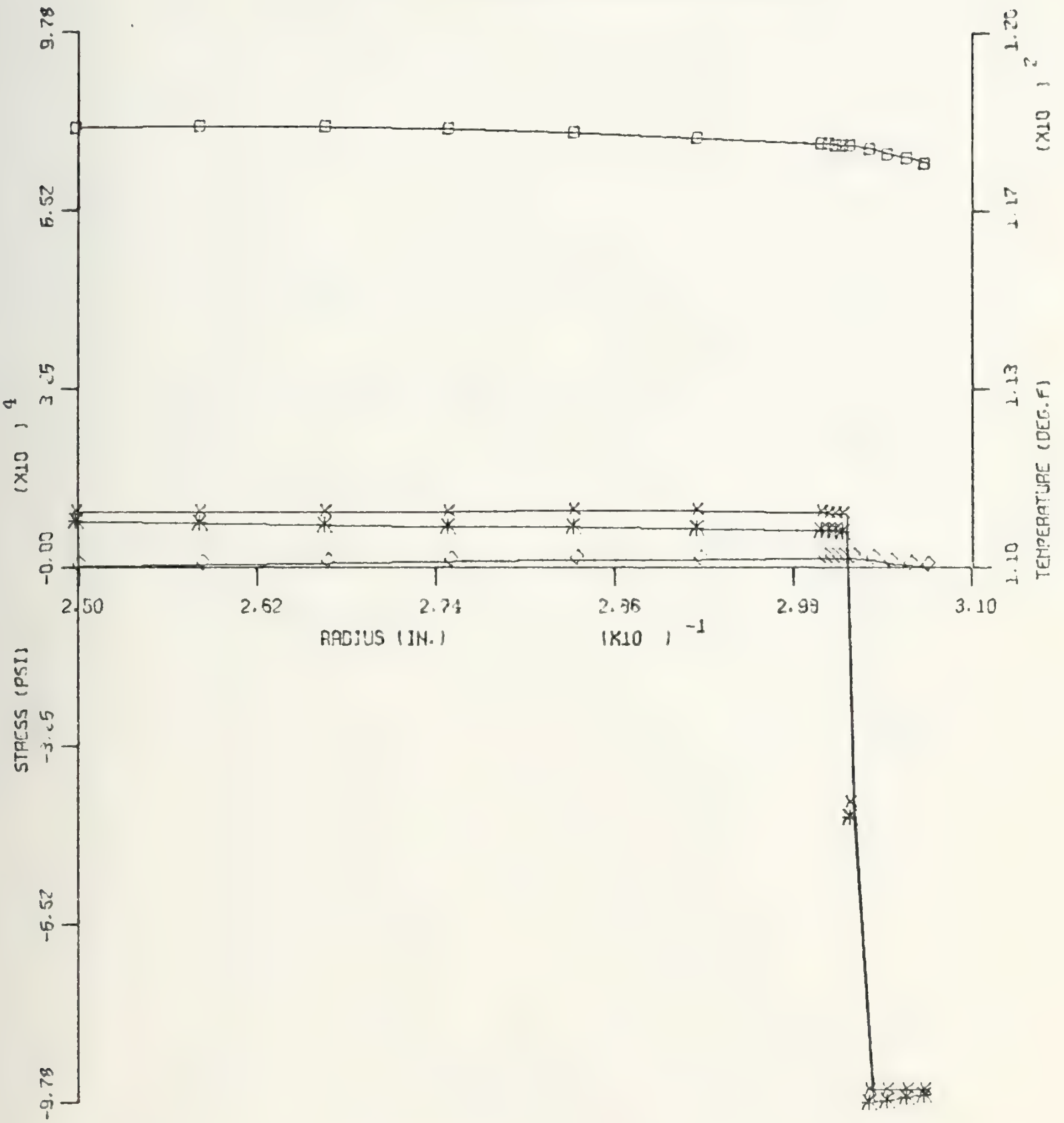
STRESS SCALE = 1.767X 10^4 PSI/INCH

TEMPERATURE SCALE = 6.7 DEGREES/INCH

RADIAL POSITION SCALE = 0.01 INCHES/INCH

FIGURE 9(i)

STRESS AND TEMPERATURE
VERSUS RADIAL POSITION
TIME = 1.650 CONF.2-60



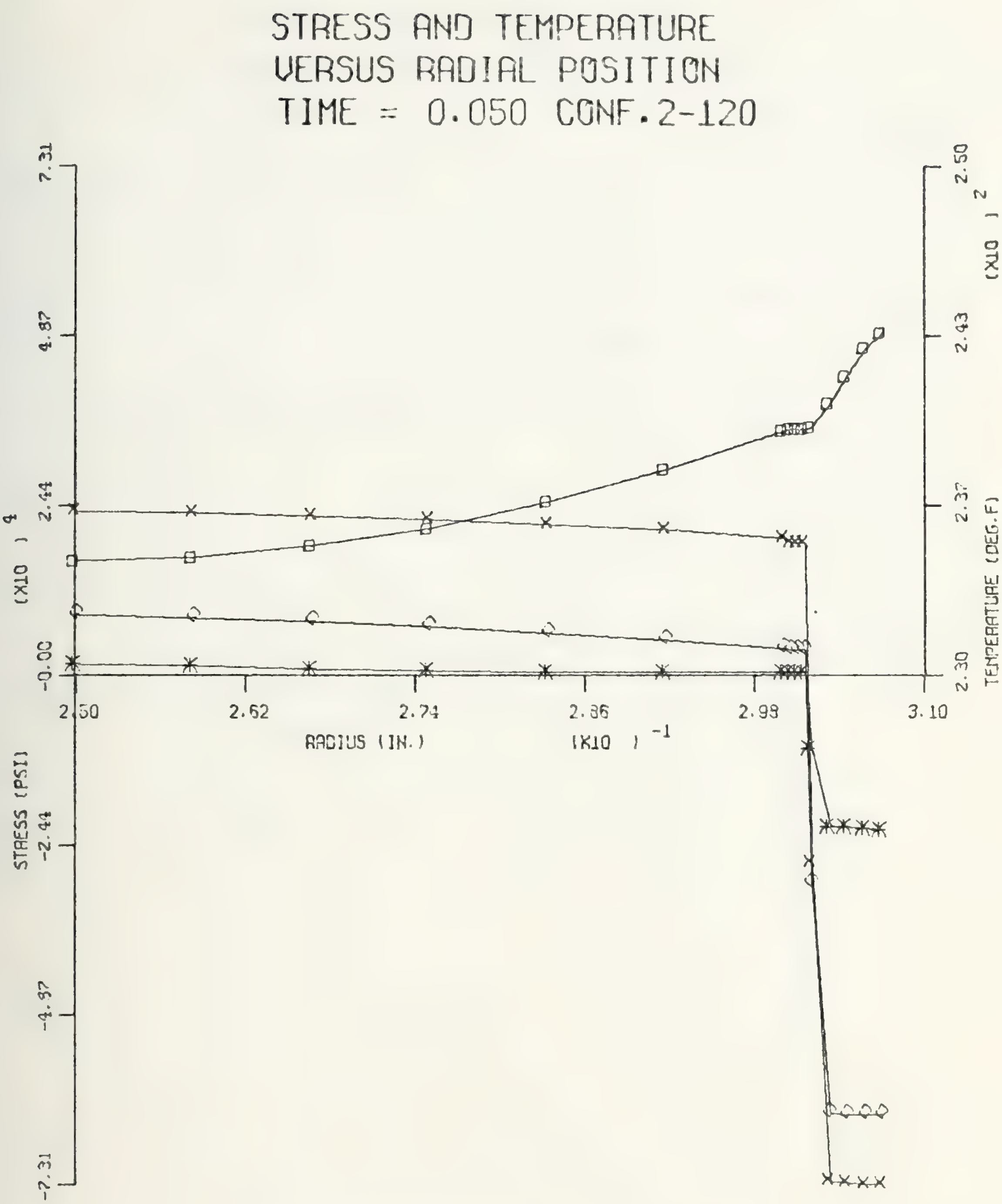
SIGMA R - DIAMOND SIGMA T - ASTERISK
SIGMA Z - CROSS TEMPERATURE - SQUARE

STRESS SCALE = 3.251 X 10^4 PSI/INCH

TEMPERATURE SCALE = 3.3 DEGREES/INCH

RADIAL POSITION SCALE = 0.01 INCHES/INCH

FIGURE 9(j)



SIGMA R - DIAMOND SIGMA T - ASTERISK
SIGMA Z - CROSS TEMPERATURE - SQUARE

STRESS SCALE = 2.435X 10^4 PSI/INCH

TEMPERATURE SCALE = 5.7 DEGREES/INCH

RADIAL POSITION SCALE = 0.01 INCHES/INCH

FIGURE 10(a)

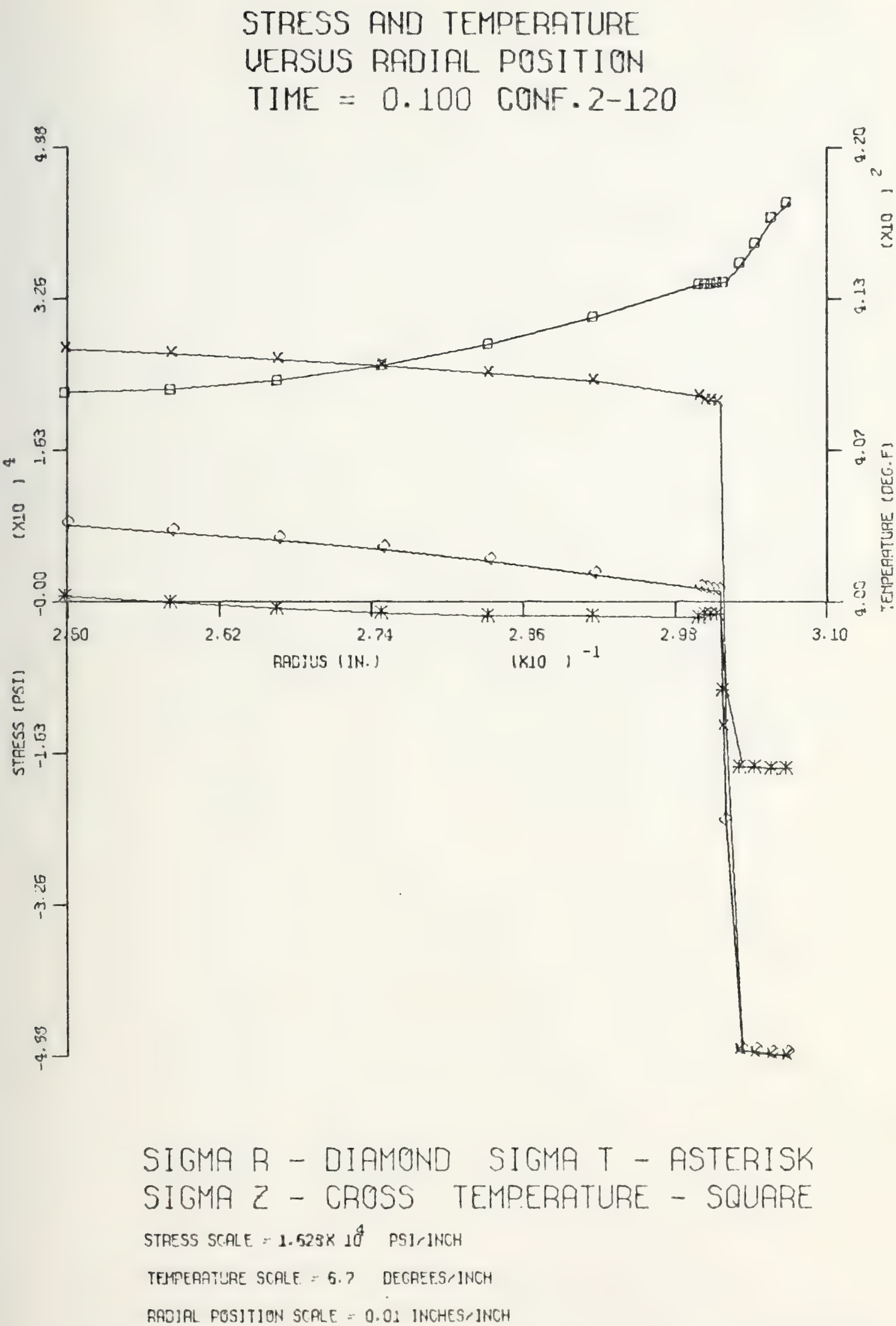
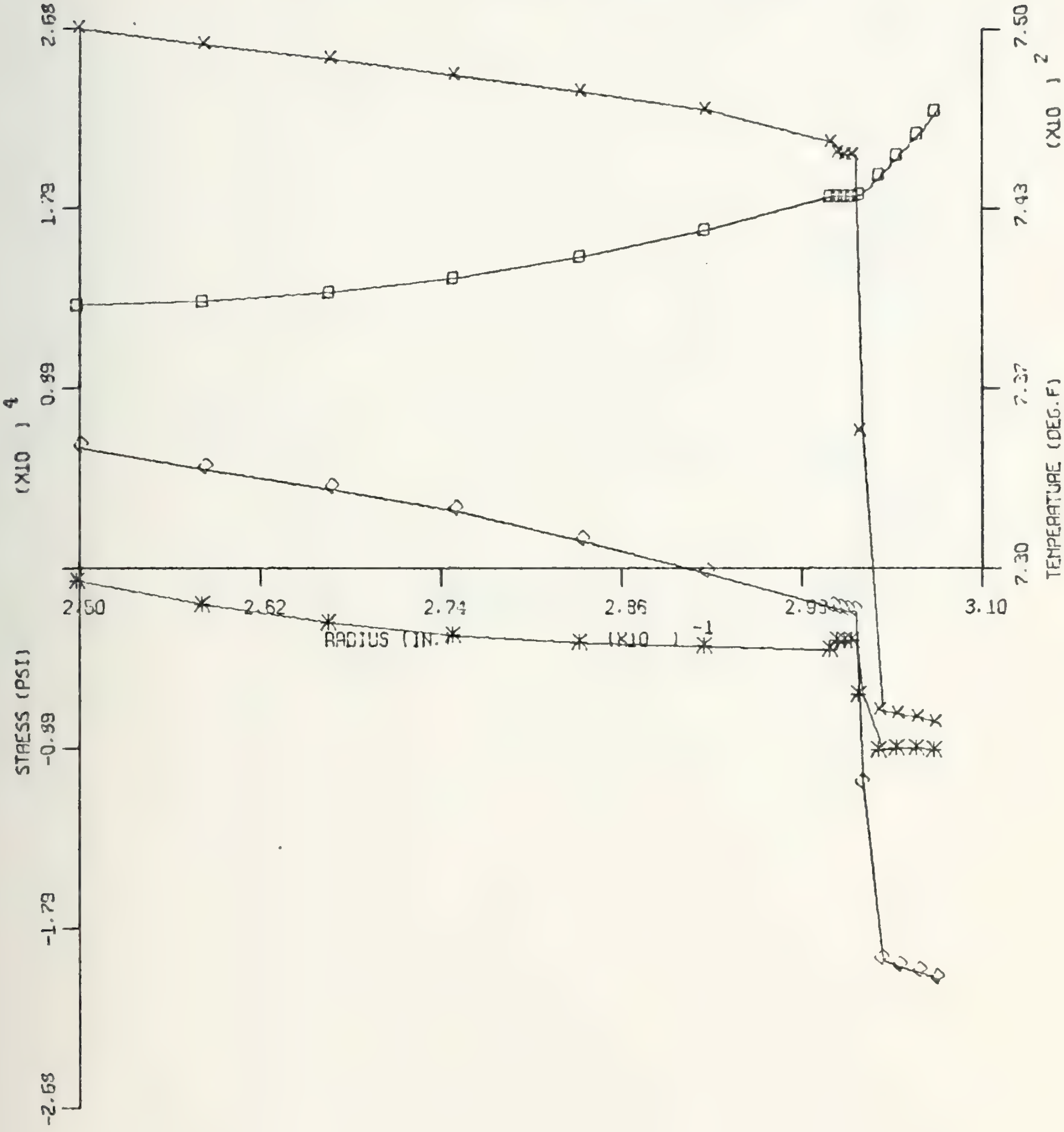

STRESS AND TEMPERATURE
VERSUS RADIAL POSITION
TIME = 0.100 CONF.2-120
STRESS (PSI)
(X10) 4
4.88
3.26
1.63
-0.00
-1.63
-3.26
-4.88
TEMPERATURE (DEG.F)
(X10) 2
4.20
4.13
4.07
4.00
2.50
2.62
2.74
2.86
2.98
3.10
RADIUS (IN.)
(X10) -1
SIGMA R - DIAMOND SIGMA T - ASTERISK
SIGMA Z - CROSS TEMPERATURE - SQUARE
STRESS SCALE = 1.628X 10 4 PSI/INCH
TEMPERATURE SCALE = 6.7 DEGREES/INCH
RADIAL POSITION SCALE = 0.01 INCHES/INCH

FIGURE 10(b)

STRESS AND TEMPERATURE
VERSUS RADIAL POSITION
TIME = 0.200 CONF.2-120



SIGMA R - DIAMOND SIGMA T - ASTERISK
SIGMA Z - CROSS TEMPERATURE - SQUARE

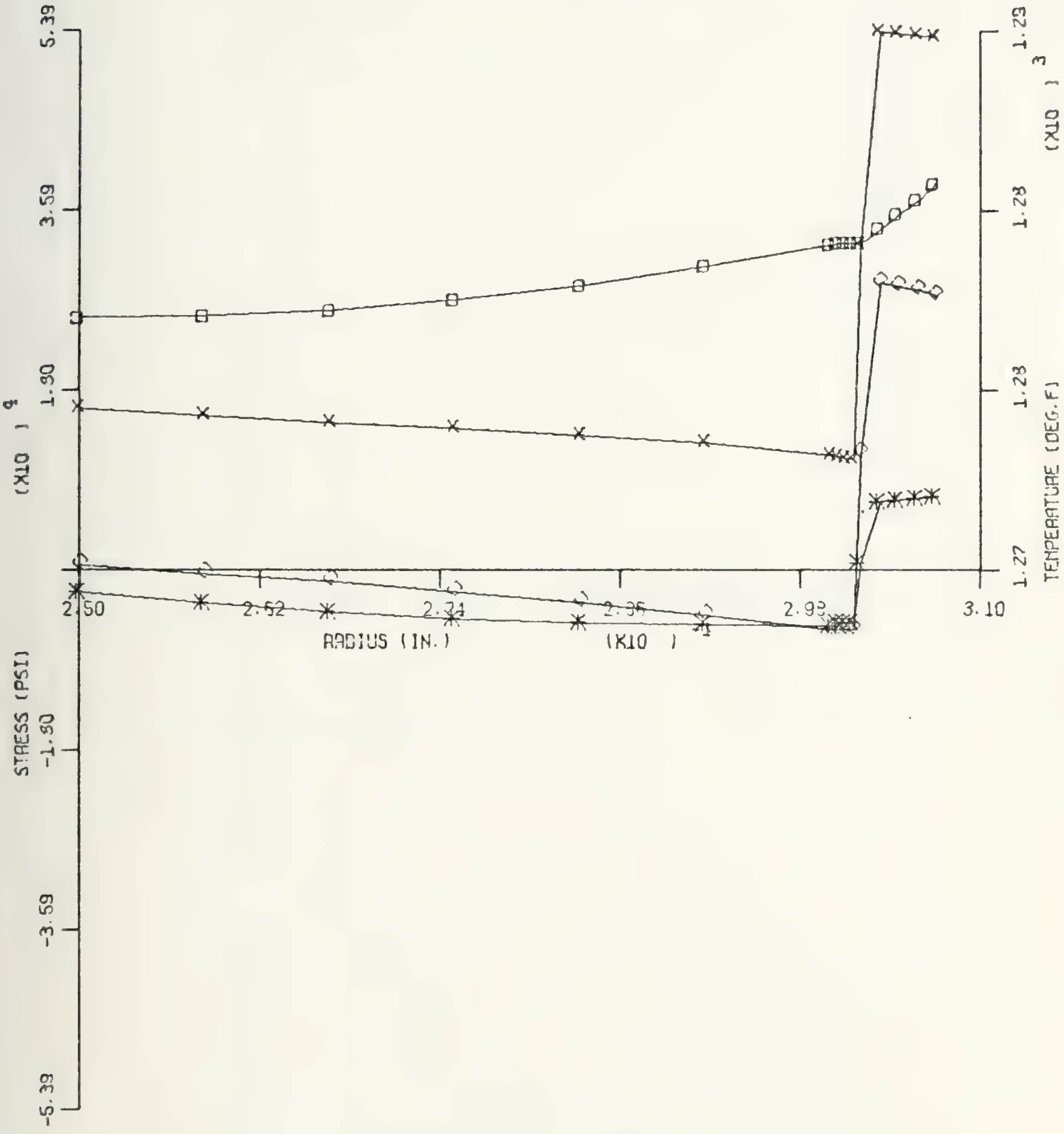
STRESS SCALE = 8.930X 10^3 PSI/INCH

TEMPERATURE SCALE = 6.7 DEGREES/INCH

RADIAL POSITION SCALE = 0.01 INCHES/INCH

FIGURE 10(c)

STRESS AND TEMPERATURE
VERSUS RADIAL POSITION
TIME = 0.400 CONF.2-120



SIGMA R - DIAMOND SIGMA T - ASTERISK
SIGMA Z - CROSS TEMPERATURE - SQUARE

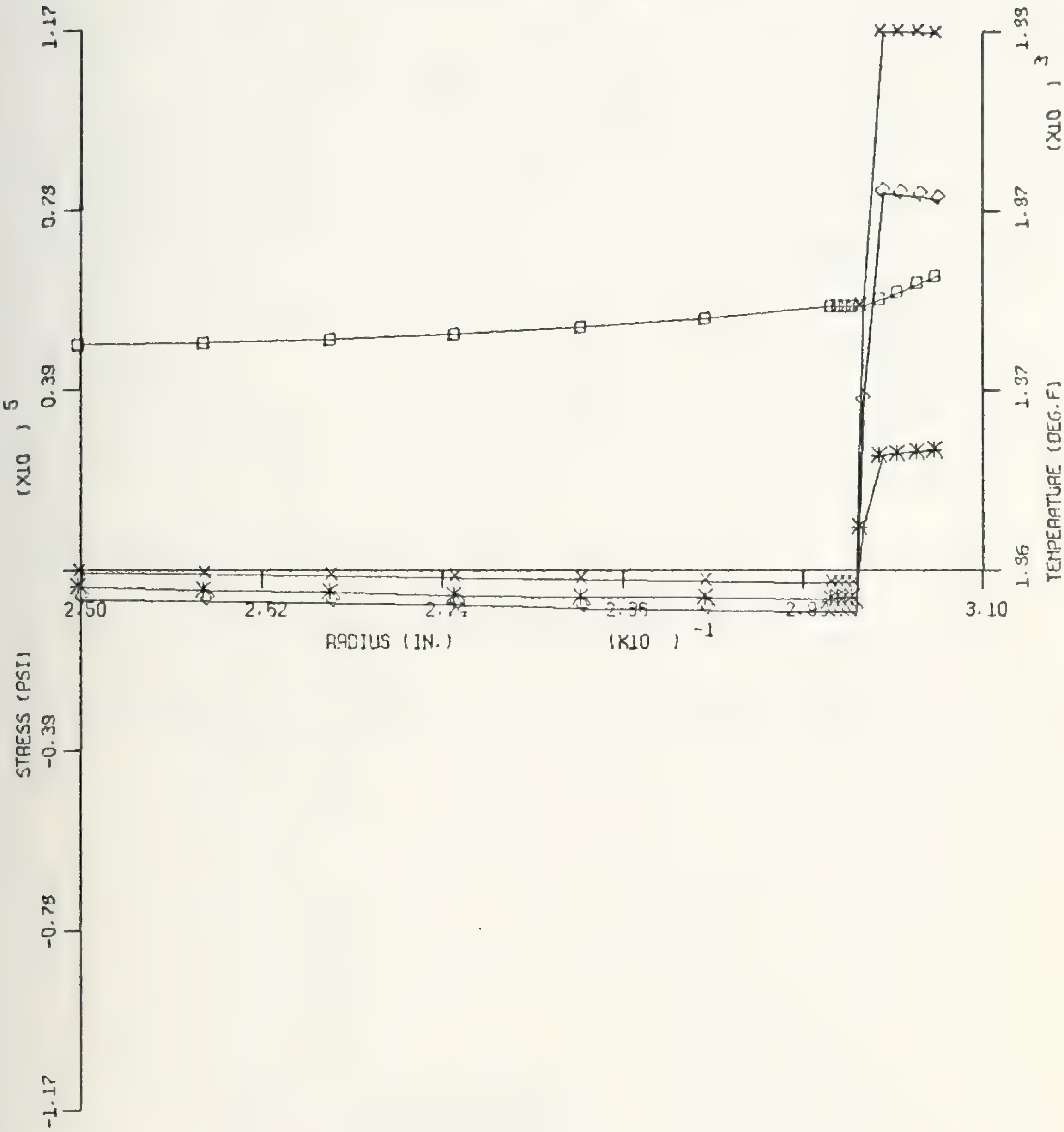
STRESS SCALE = 1.796 X 10^4 PSI/INCH

TEMPERATURE SCALE = 6.7 DEGREES/INCH

RADIAL POSITION SCALE = 0.01 INCHES/INCH

FIGURE 10(d)

STRESS AND TEMPERATURE
VERSUS RADIAL POSITION
TIME = 0.750 CONF.2-120



SIGMA R - DIAMOND SIGMA T - ASTERISK
SIGMA Z - CROSS TEMPERATURE - SQUARE

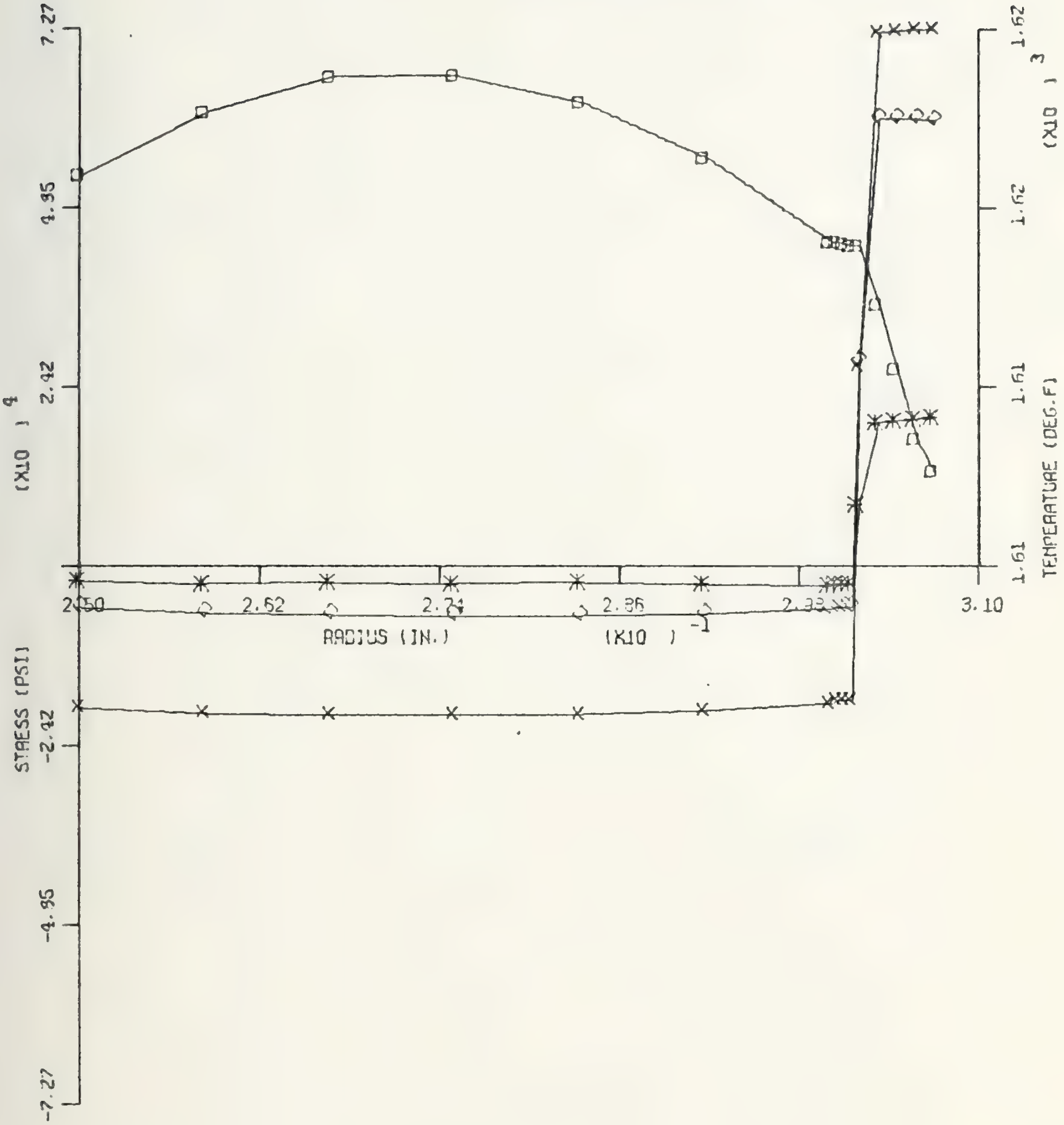
STRESS SCALE = 3.910X 10^4 PSI/INCH

TEMPERATURE SCALE = 6.7 DEGREES/INCH

RADIAL POSITION SCALE = 0.01 INCHES/INCH

FIGURE 10(e)

STRESS AND TEMPERATURE
VERSUS RADIAL POSITION
TIME = 0.800 CONF.2-120



SIGMA R - DIAMOND SIGMA T - ASTERISK
SIGMA Z - CROSS TEMPERATURE - SQUARE

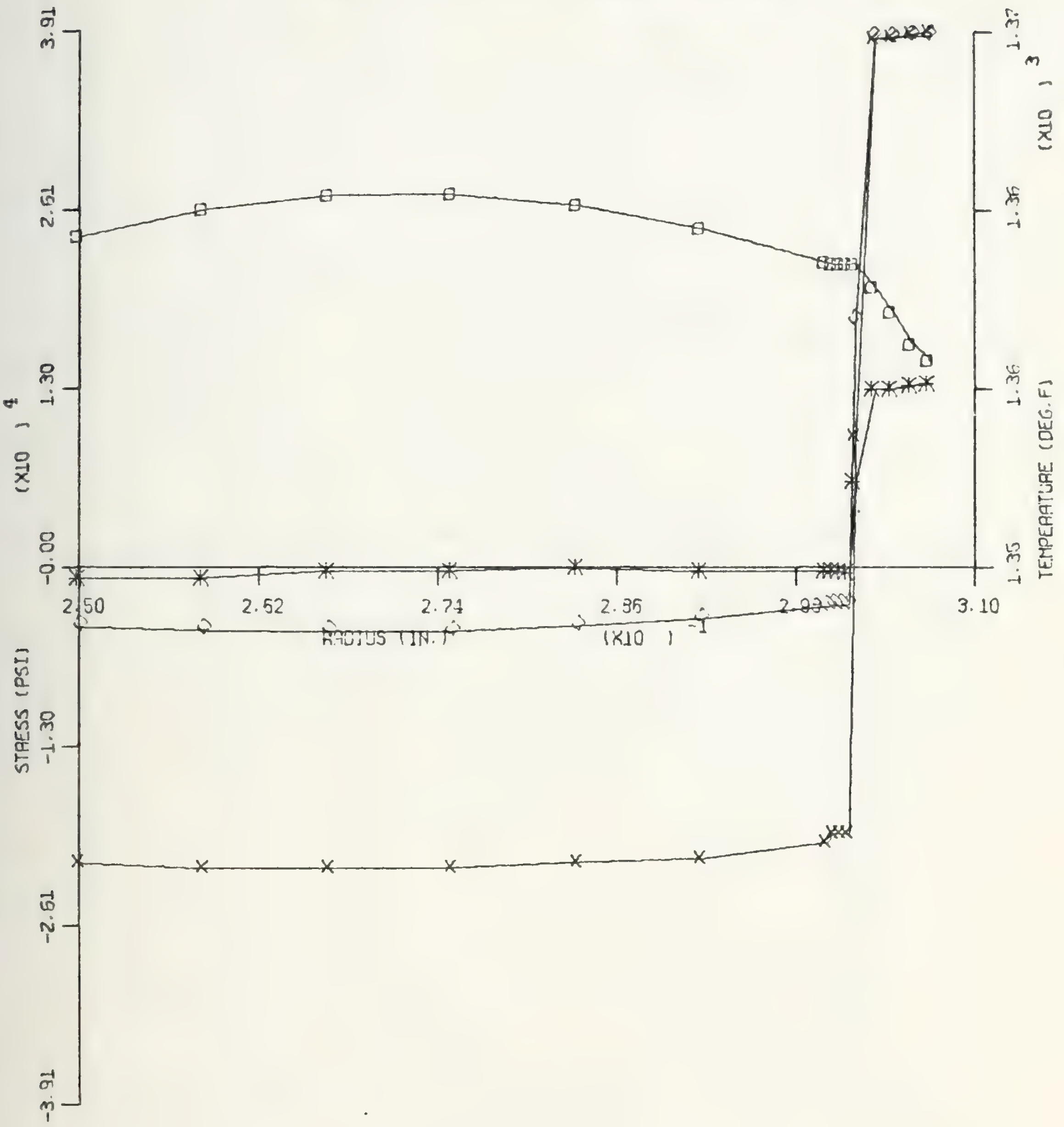
STRESS SCALE = 2.424X 10^{4} PSI/INCH

TEMPERATURE SCALE = 3.3 DEGREES/INCH

RADIAL POSITION SCALE = 0.01 INCHES/INCH

FIGURE 10(f)

STRESS AND TEMPERATURE
VERSUS RADIAL POSITION
TIME = 0.850 CONF.2-120



SIGMA R - DIAMOND SIGMA T - ASTERISK
SIGMA Z - CROSS TEMPERATURE - SQUARE

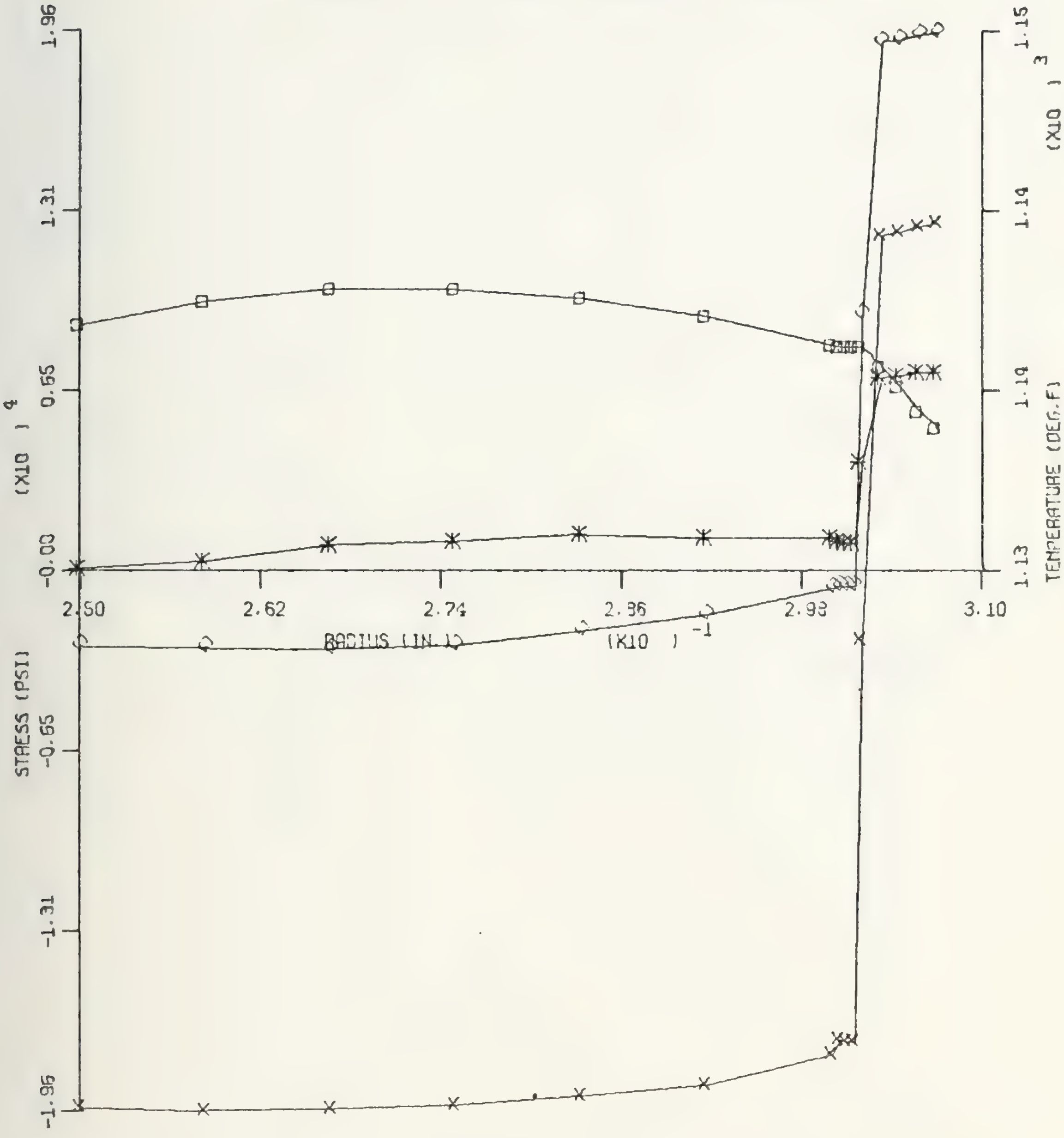
STRESS SCALE = 1.304 X 10[3] PSI/INCH

TEMPERATURE SCALE = 6.7 DEGREES/INCH

RADIAL POSITION SCALE = 0.01 INCHES/INCH

FIGURE 10(g)

STRESS AND TEMPERATURE
VERSUS RADIAL POSITION
TIME = 0.900 CONF.2-120



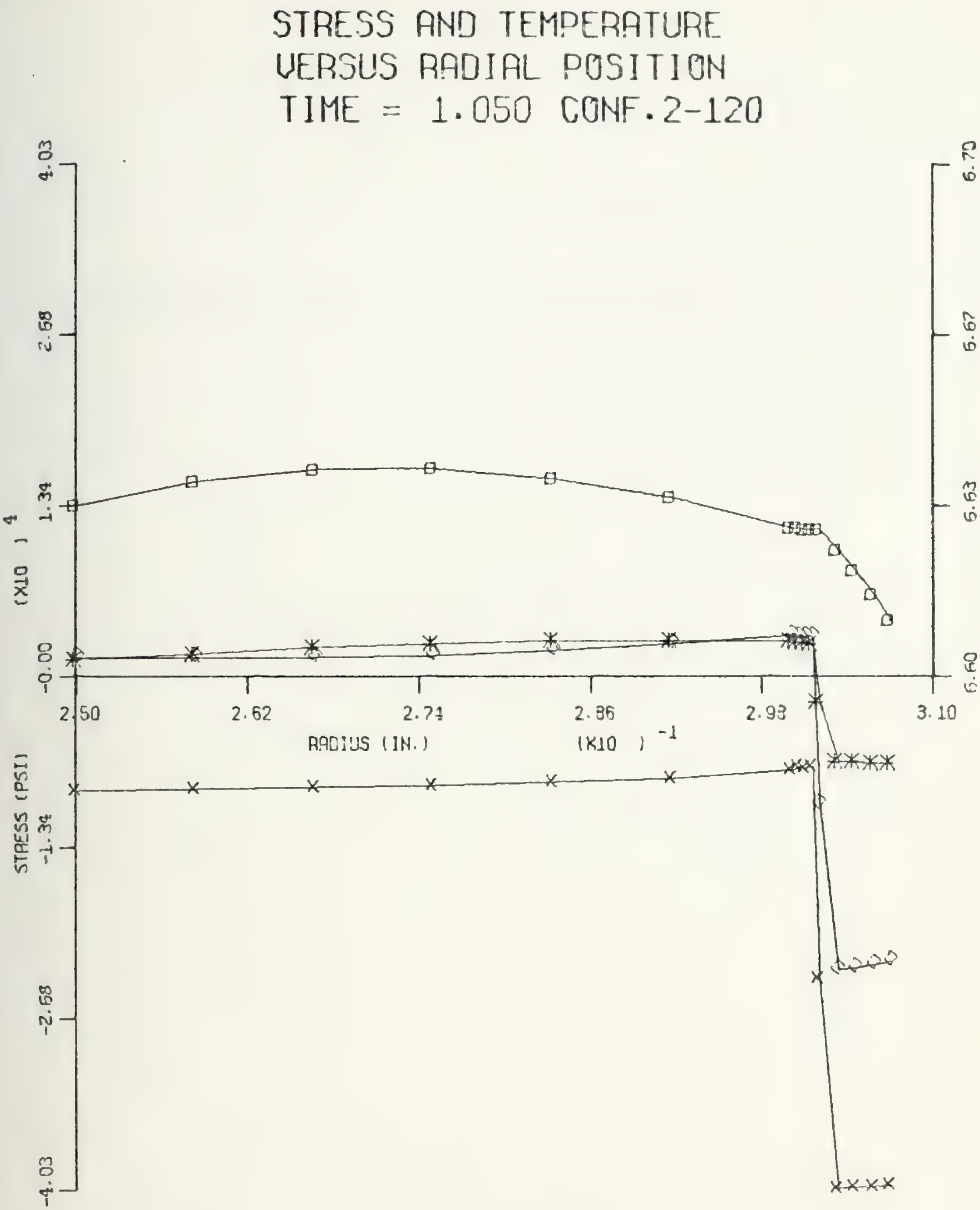
SIGMA R - DIAMOND SIGMA T - ASTERISK
SIGMA Z - CROSS TEMPERATURE - SQUARE

STRESS SCALE = 5.537X 10^3 PSI/INCH

TEMPERATURE SCALE = 6.7 DEGREES/INCH

RADIAL POSITION SCALE = 0.01 INCHES/INCH

FIGURE 10(h)



SIGMA R - DIAMOND SIGMA T - ASTERISK
SIGMA Z - CROSS TEMPERATURE - SQUARE

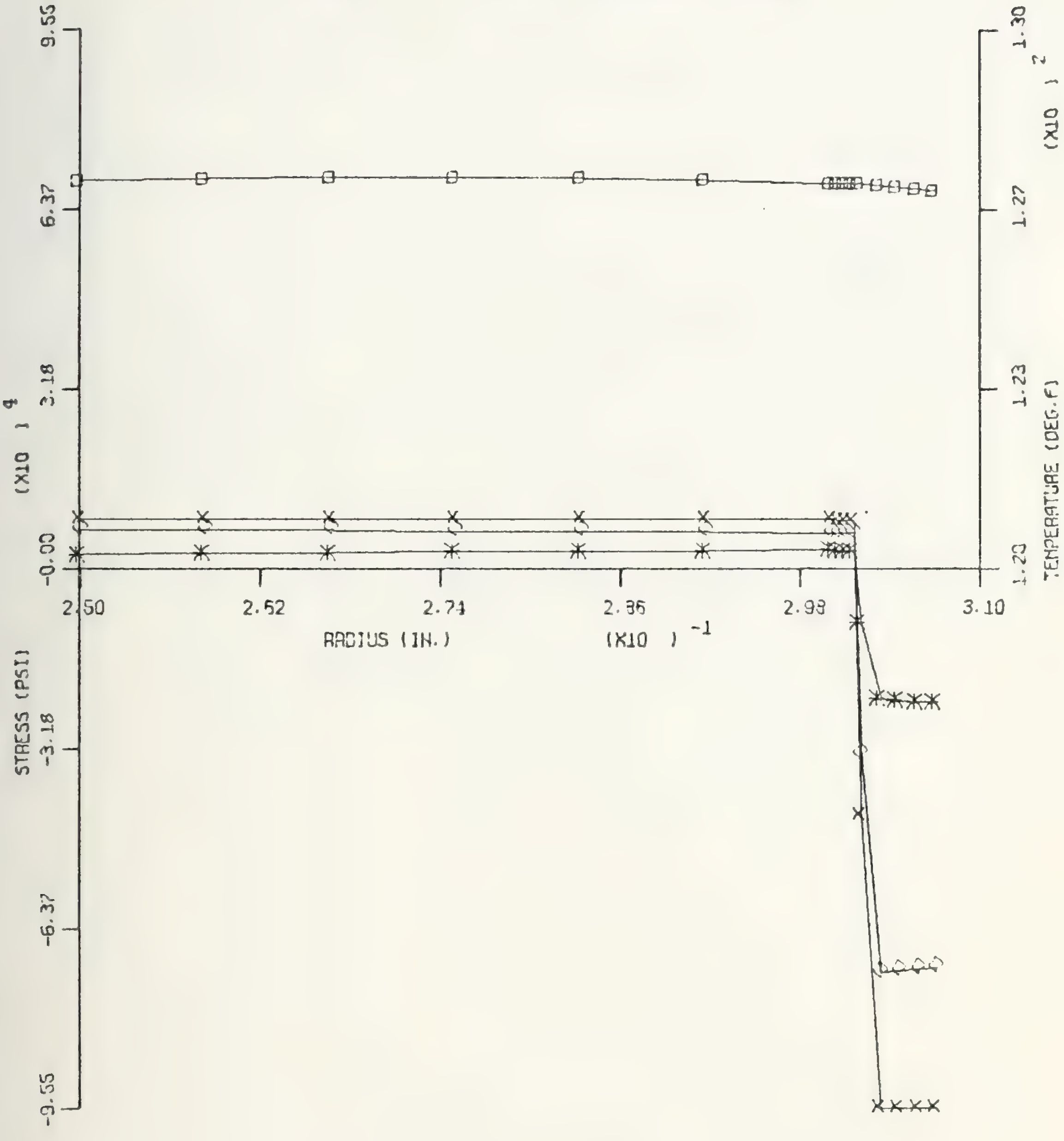
STRESS SCALE = 1.342X 10⁴ PSI/INCH

TEMPERATURE SCALE = 3.3 DEGREES/INCH

RADIAL POSITION SCALE = 0.01 INCHES/INCH

FIGURE 10(i)

STRESS AND TEMPERATURE
VERSUS RADIAL POSITION
TIME = 1.650 CONF.2-120



SIGMA R - DIAMOND SIGMA T - ASTERISK
SIGMA Z - CROSS TEMPERATURE - SQUARE

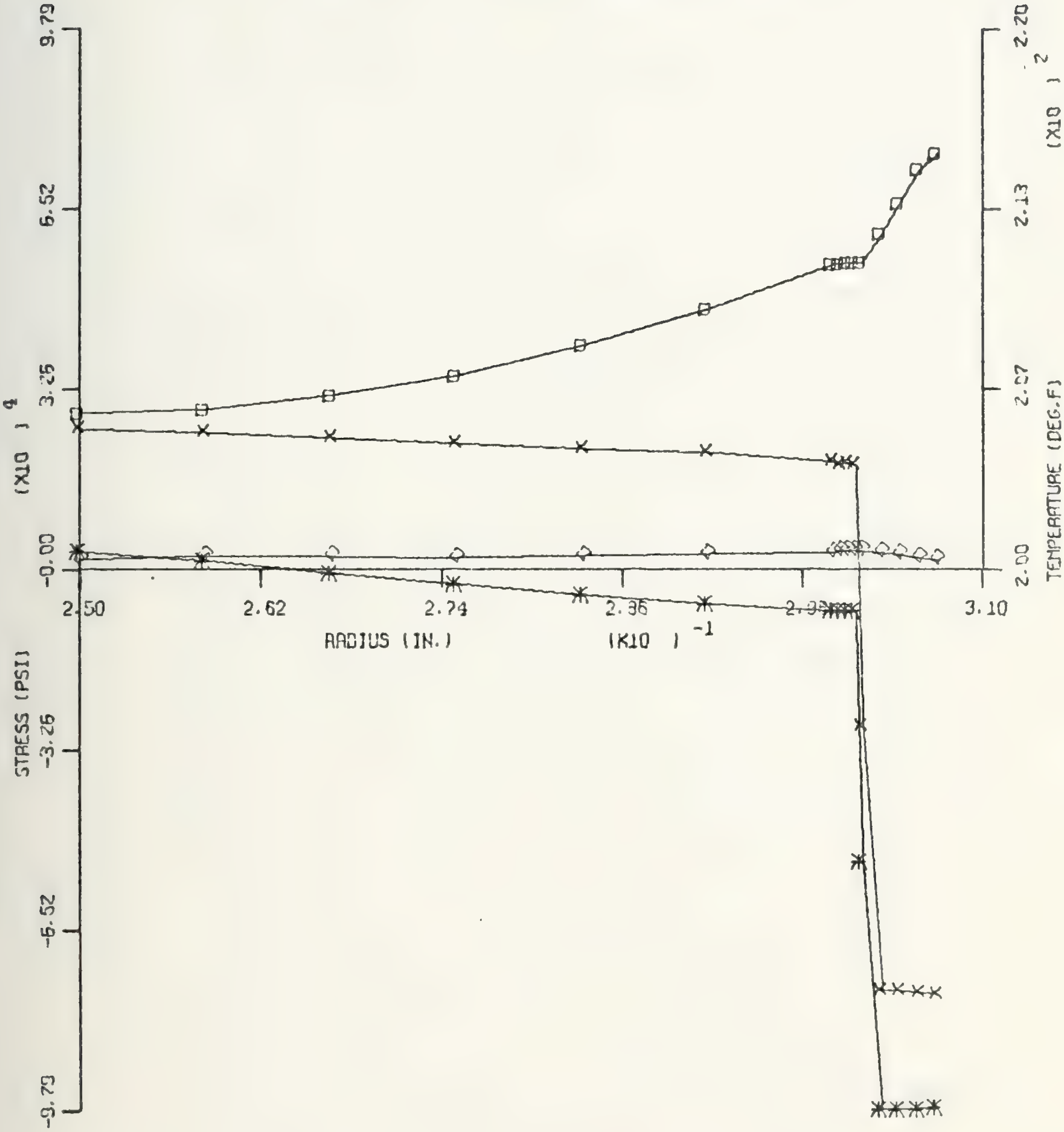
STRESS SCALE = 3.183K 10^4 PSI/INCH

TEMPERATURE SCALE = 3.3 DEGREES/INCH

RADIAL POSITION SCALE = 0.01 INCHES/INCH

FIGURE 10(j)

STRESS AND TEMPERATURE
VERSUS RADIAL POSITION
TIME = 0.050 CONF.2-180



SIGMA R - DIAMOND SIGMA T - ASTERISK
SIGMA Z - CROSS TEMPERATURE - SQUARE

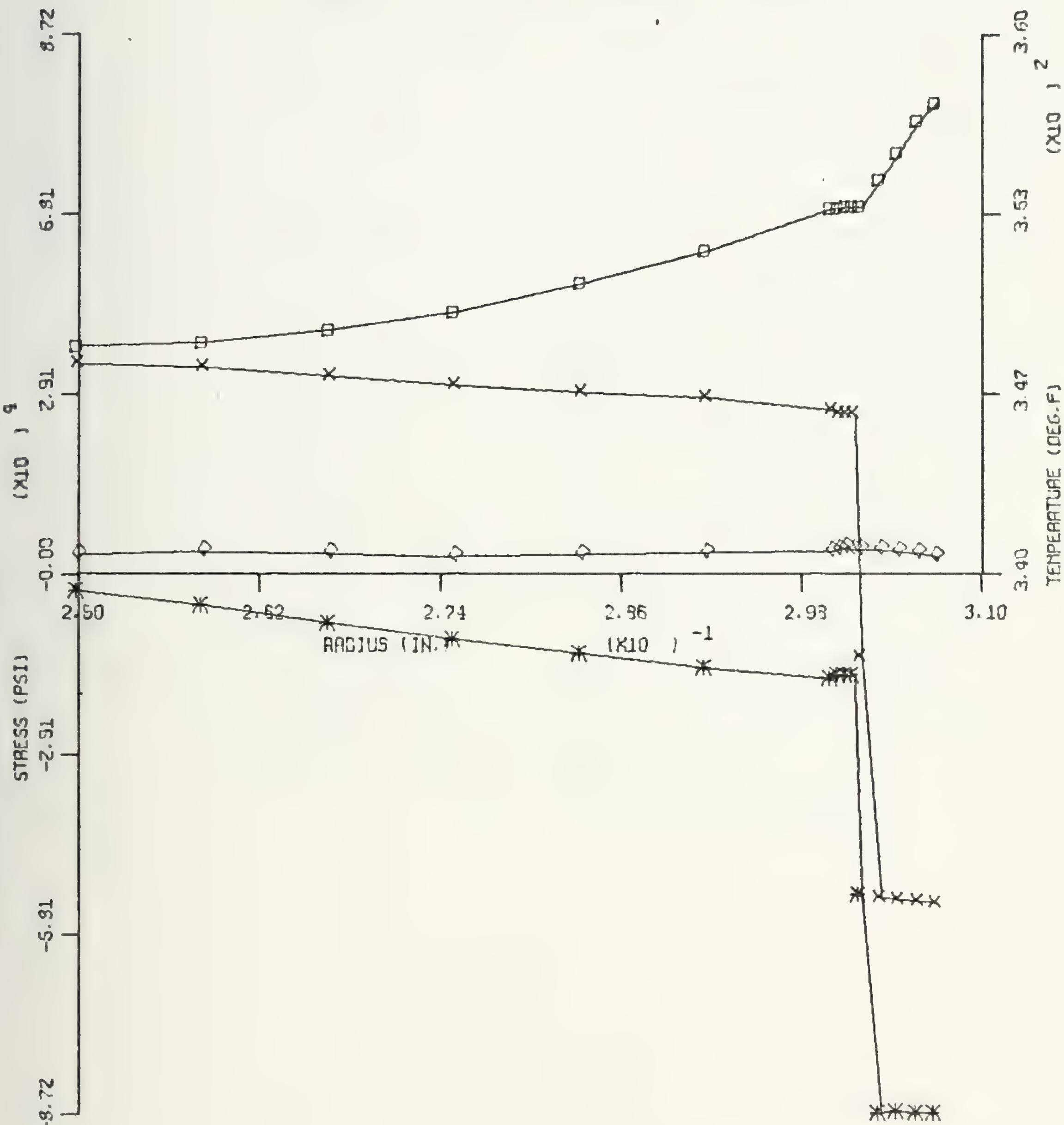
STRESS SCALE = 3.262X 10^4 PSI/INCH

TEMPERATURE SCALE = 6.7 DEGREES/INCH

RADIAL POSITION SCALE = 0.01 INCHES/INCH

FIGURE 11(a)

STRESS AND TEMPERATURE
VERSUS RADIAL POSITION
TIME = 0.100 CONF.2-180



SIGMA R - DIAMOND SIGMA T - ASTERISK
SIGMA Z - CROSS TEMPERATURE - SQUARE

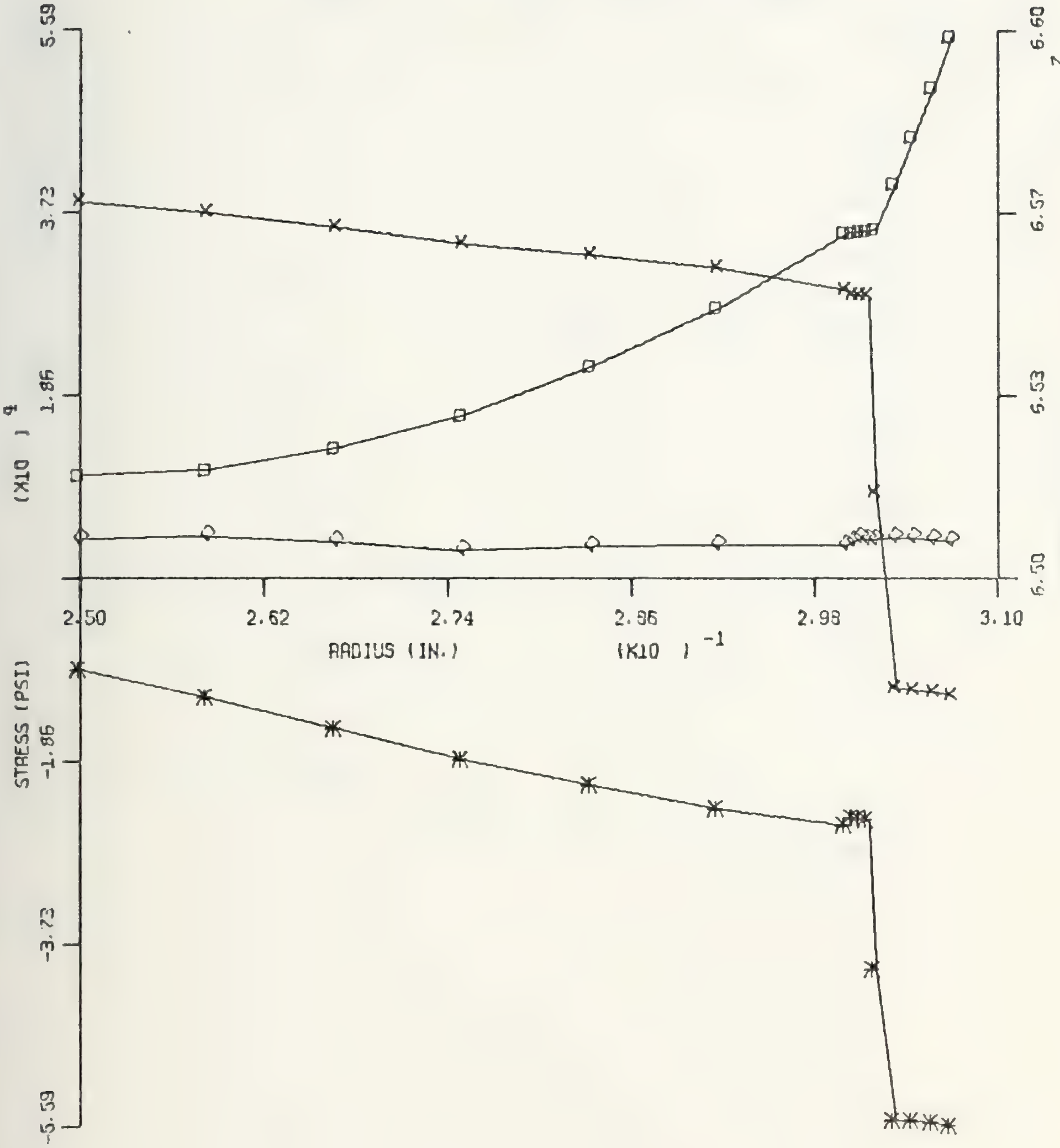
STRESS SCALE = 2.907X 10^4 PSI/INCH

TEMPERATURE SCALE = 6.7 DEGREES/INCH

RADIAL POSITION SCALE = 0.01 INCHES/INCH

FIGURE 11(b)

STRESS AND TEMPERATURE
VERSUS RADIAL POSITION
TIME = 0.200 CONF.2-180



SIGMA R - DIAMOND SIGMA T - ASTERISK
SIGMA Z - CROSS TEMPERATURE - SQUARE

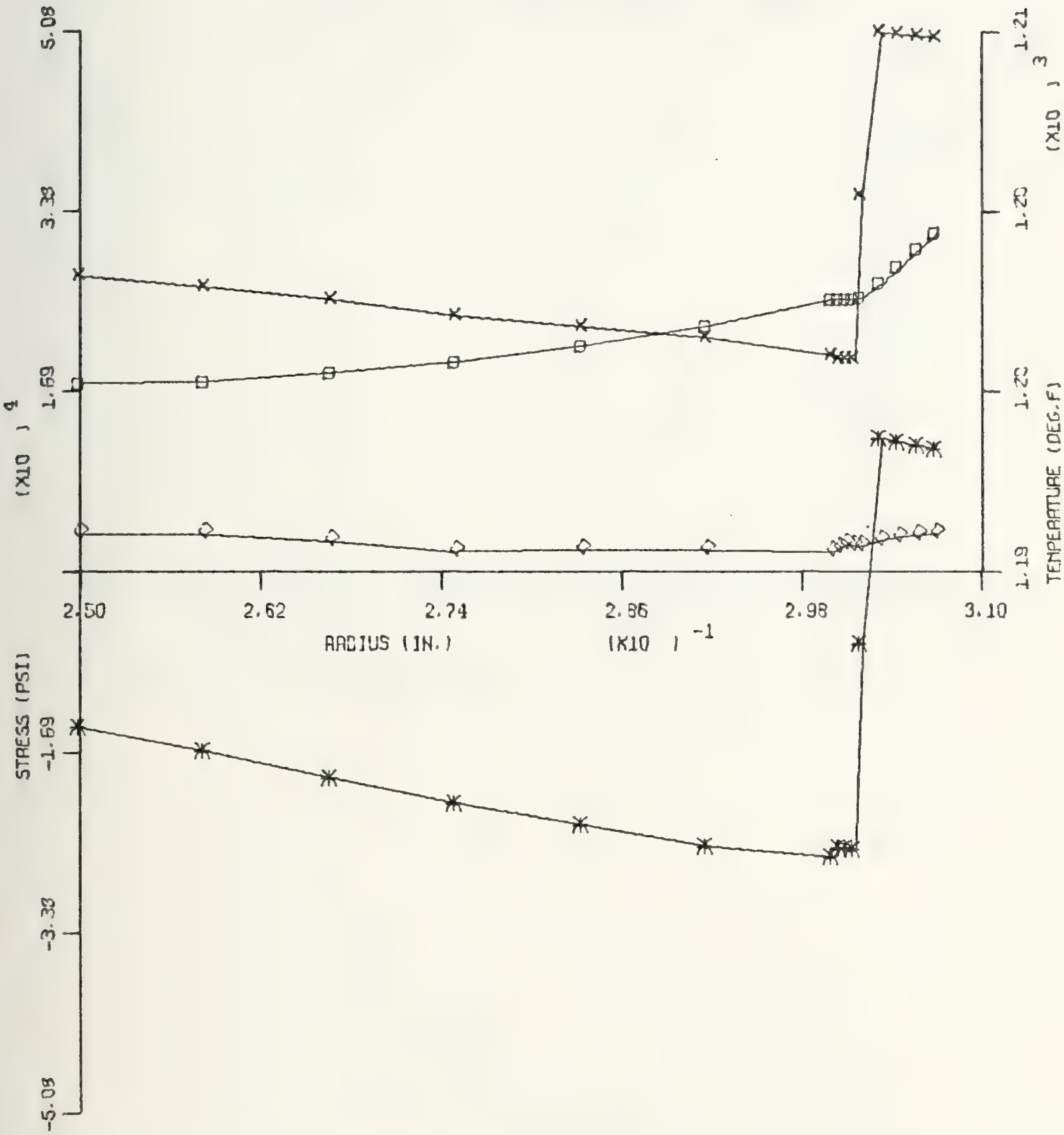
STRESS SCALE = 1.863X 10^4 PSI/INCH

TEMPERATURE SCALE = 3.3 DEGREES/INCH

RADIAL POSITION SCALE = 0.01 INCHES/INCH

FIGURE 11(c)

STRESS AND TEMPERATURE
VERSUS RADIAL POSITION
TIME = 0.400 CONF.2-180



SIGMA R - DIAMOND SIGMA T - ASTERISK
SIGMA Z - CROSS TEMPERATURE - SQUARE

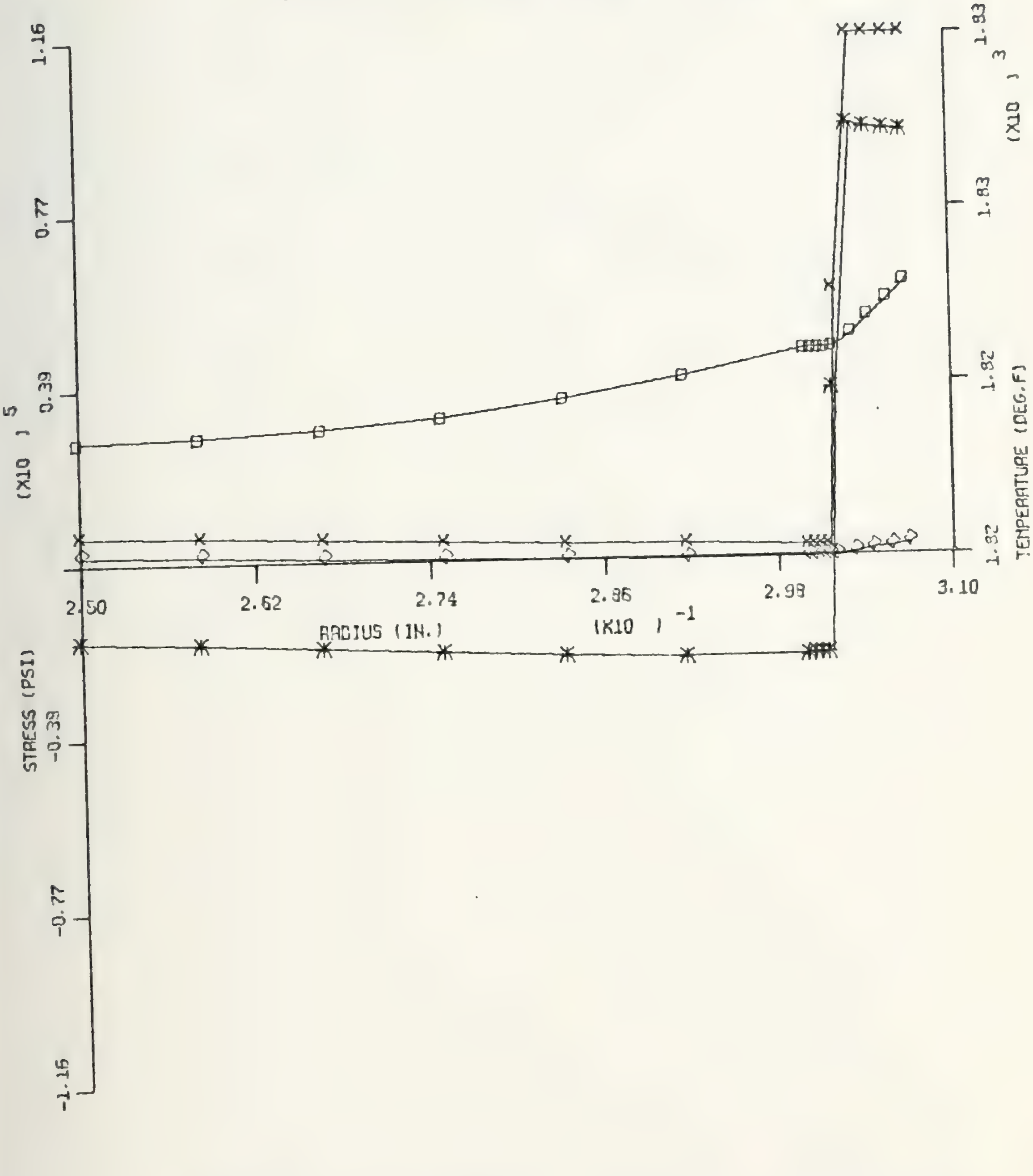
STRESS SCALE = 1.692×10^4 PSI/INCH

TEMPERATURE SCALE = 6.7 DEGREES/INCH

RADIAL POSITION SCALE = 0.01 INCHES/INCH

FIGURE 11(d)

STRESS AND TEMPERATURE
VERSUS RADIAL POSITION
TIME = 0.750 CONF.2-180



SIGMA R - DIAMOND SIGMA T - ASTERISK
SIGMA Z - CROSS TEMPERATURE - SQUARE

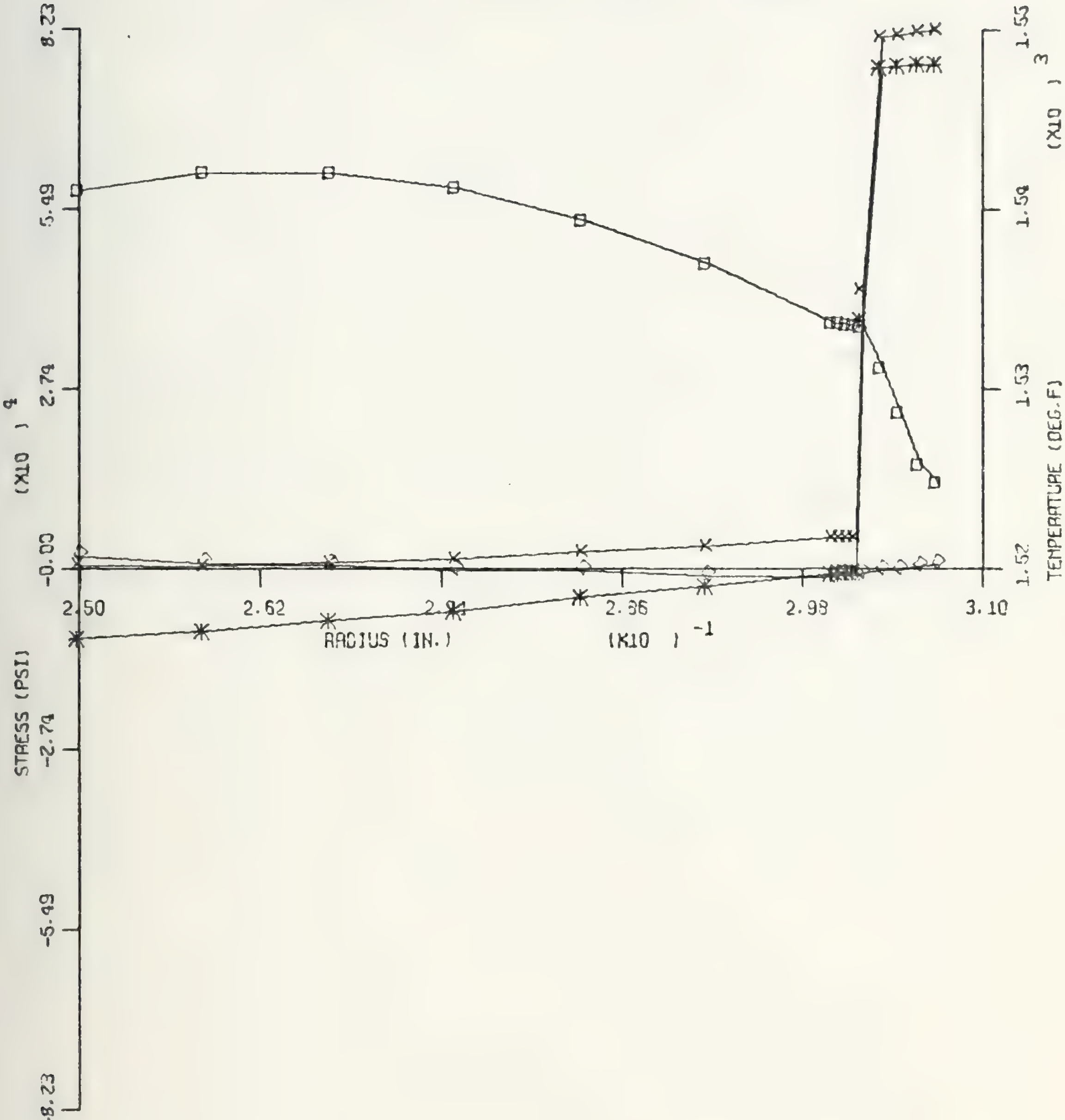
STRESS SCALE = 3.854X 10^{4} PSI/INCH

TEMPERATURE SCALE = 3.3 DEGREES/INCH

RADIAL POSITION SCALE = 0.01 INCHES/INCH

FIGURE 11(e)

STRESS AND TEMPERATURE
VERSUS RADIAL POSITION
TIME = 0.800 CONF.2-180



SIGMA R - DIAMOND SIGMA T - ASTERISK
SIGMA Z - CROSS TEMPERATURE - SQUARE

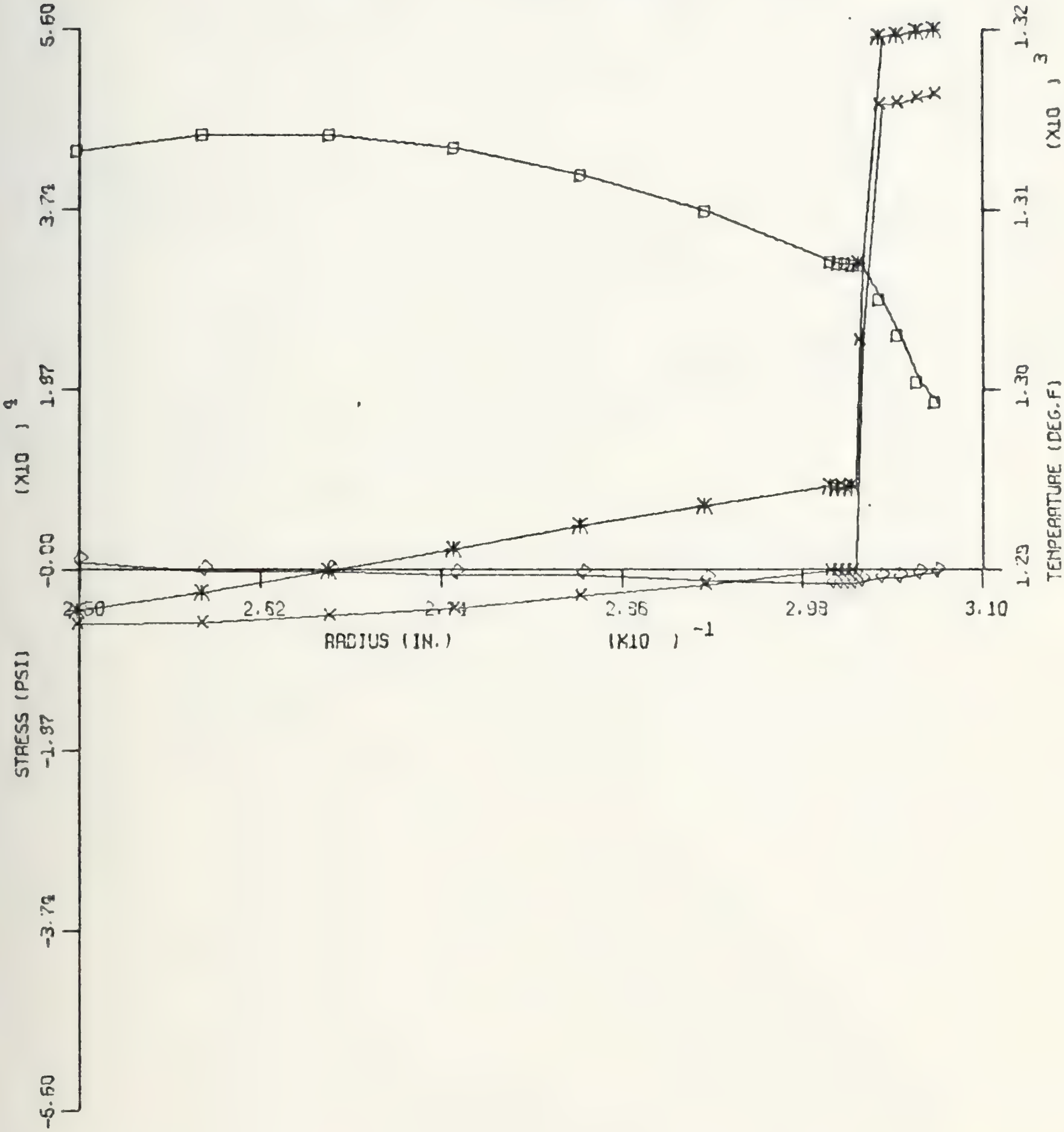
STRESS SCALE = 2.744X 10^4 PSI/INCH

TEMPERATURE SCALE = 10.0 DEGREES/INCH

RADIAL POSITION SCALE = 0.01 INCHES/INCH

FIGURE 11(f)

STRESS AND TEMPERATURE
VERSUS RADIAL POSITION
TIME = 0.850 CONF.2-180



SIGMA R - DIAMOND SIGMA T - ASTERISK
SIGMA Z - CROSS TEMPERATURE - SQUARE

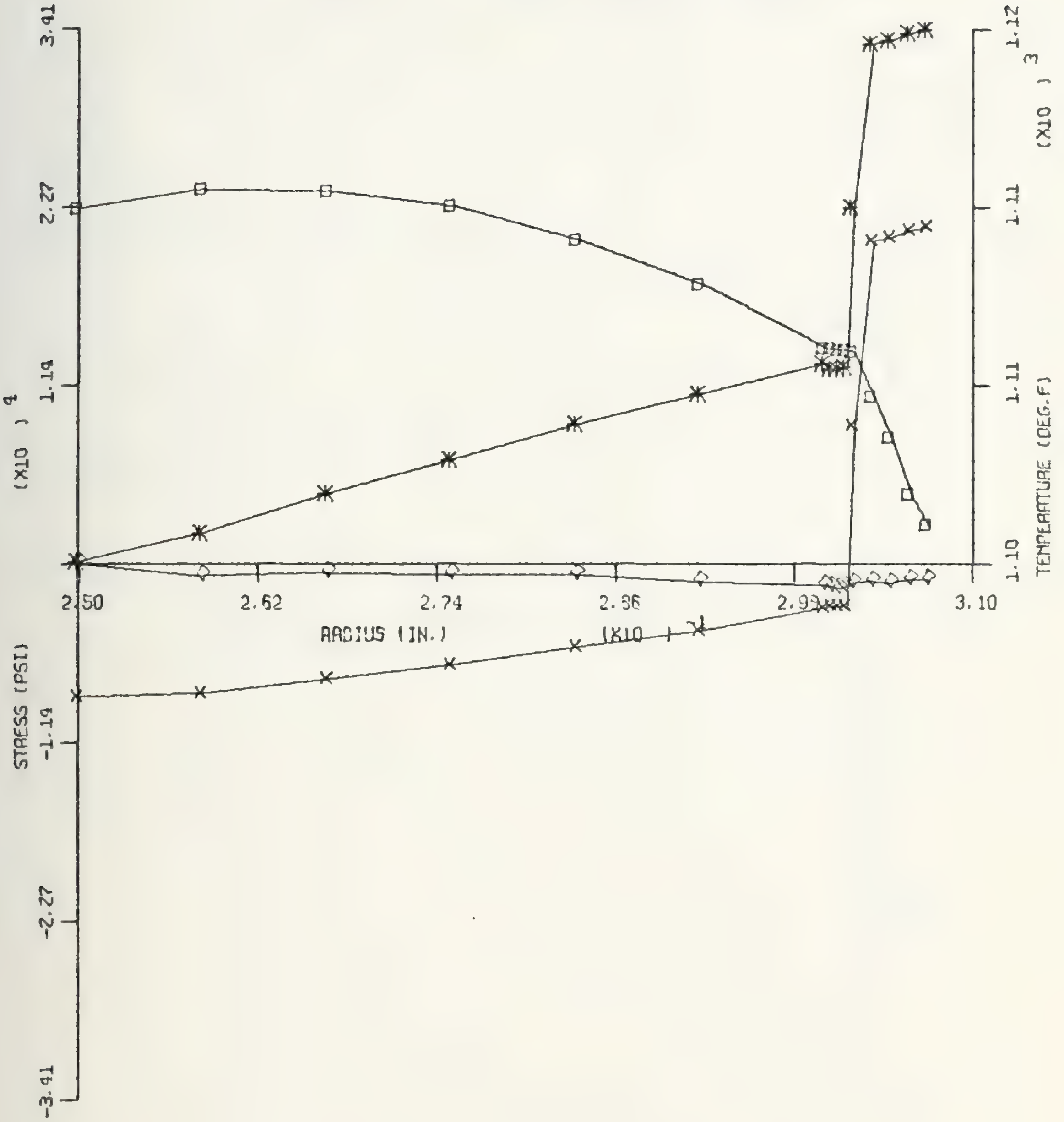
STRESS SCALE = 1.866X 10 4 PSI/INCH

TEMPERATURE SCALE = 10.0 DEGREES/INCH

RADIAL POSITION SCALE = 0.01 INCHES/INCH

FIGURE 11(g)

STRESS AND TEMPERATURE
VERSUS RADIAL POSITION
TIME = 0.900 CONF.2-180



SIGMA R - DIAMOND SIGMA T - ASTERISK
SIGMA Z - CROSS TEMPERATURE - SQUARE

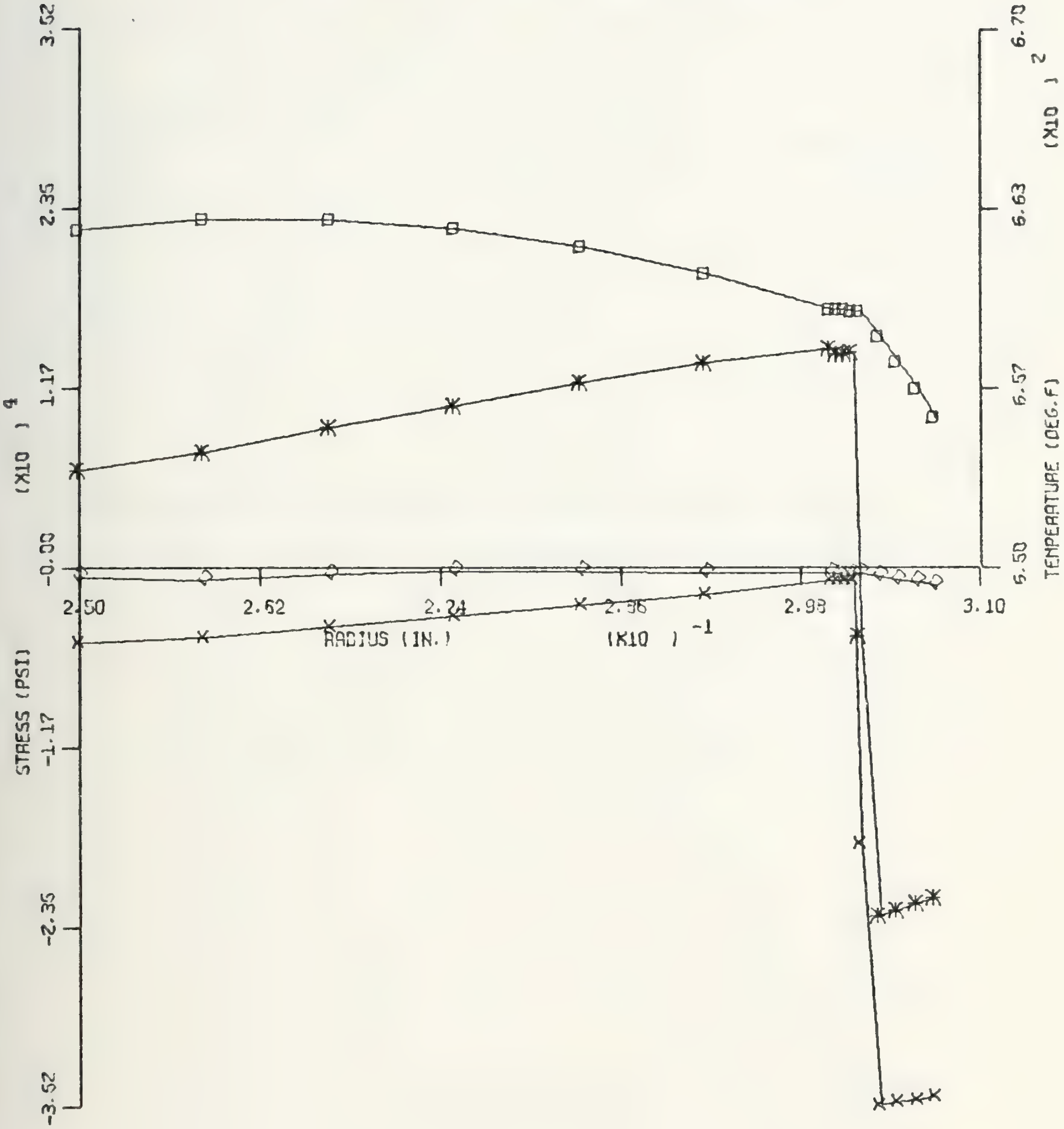
STRESS SCALE = 1.137X 10^4 PSI/INCH

TEMPERATURE SCALE = 6.7 DEGREES/INCH

RADIAL POSITION SCALE = 0.01 INCHES/INCH

FIGURE 11(h)

STRESS AND TEMPERATURE
VERSUS RADIAL POSITION
TIME = 1.050 CONF.2-180



SIGMA R - DIAMOND SIGMA T - ASTERISK
SIGMA Z - CROSS TEMPERATURE - SQUARE

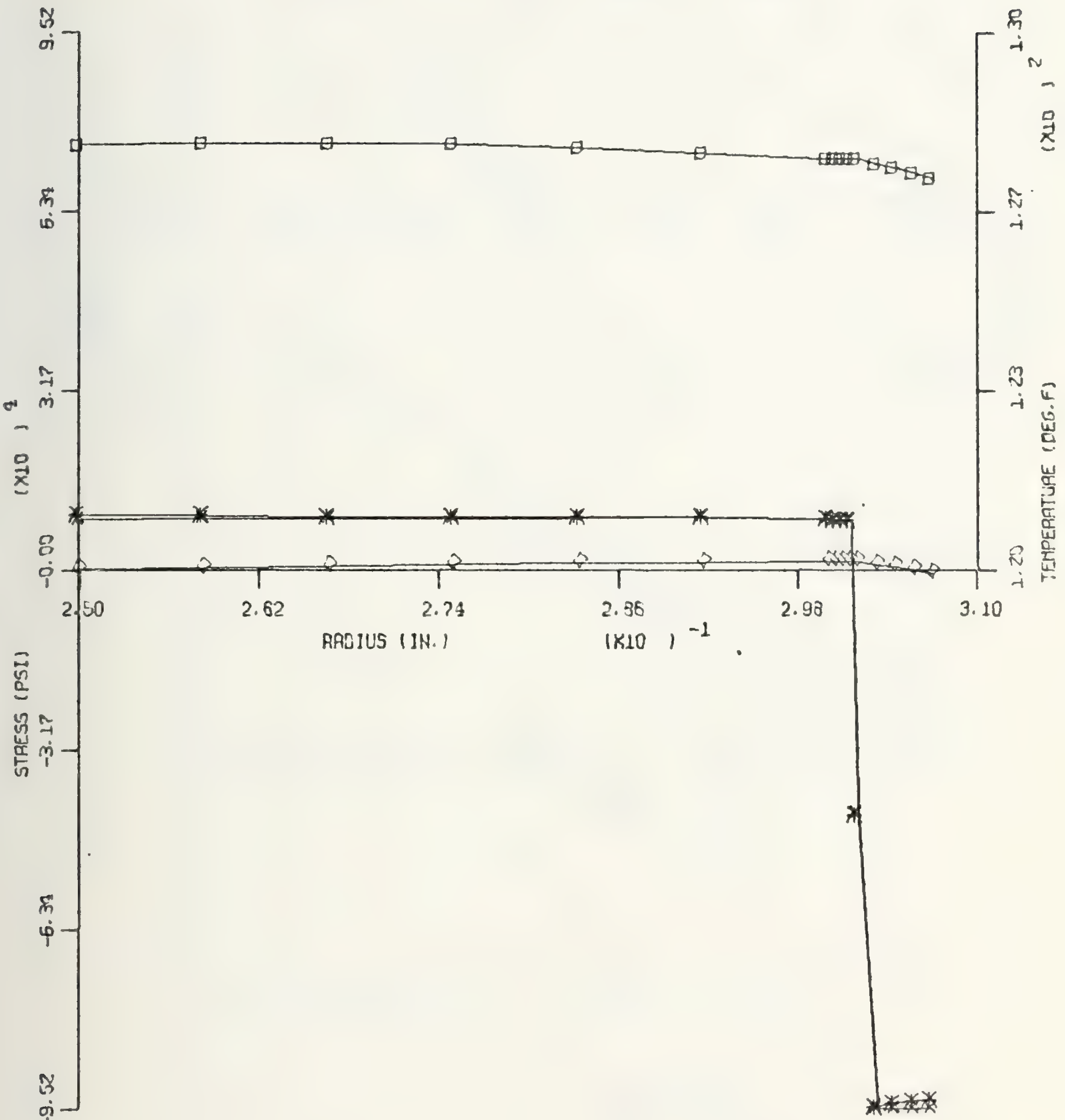
STRESS SCALE = 1.173X 10^4 PSI/INCH

TEMPERATURE SCALE = 6.7 DEGREES/INCH

RADIAL POSITION SCALE = 0.01 INCHES/INCH

FIGURE 11(i)

STRESS AND TEMPERATURE
VERSUS RADIAL POSITION
TIME = 1.650 CONF.2-180



SIGMA R - DIAMOND SIGMA T - ASTERISK
SIGMA Z - CROSS TEMPERATURE - SQUARE

STRESS SCALE = 3.172X 10^{4} PSI/INCH

TEMPERATURE SCALE = 3.3 DEGREES/INCH

RADIAL POSITION SCALE = 0.01 INCHES/INCH

FIGURE 11(j)

STRESS AND TEMPERATURE
VERSUS
CIRCUMFERENTIAL POSITION

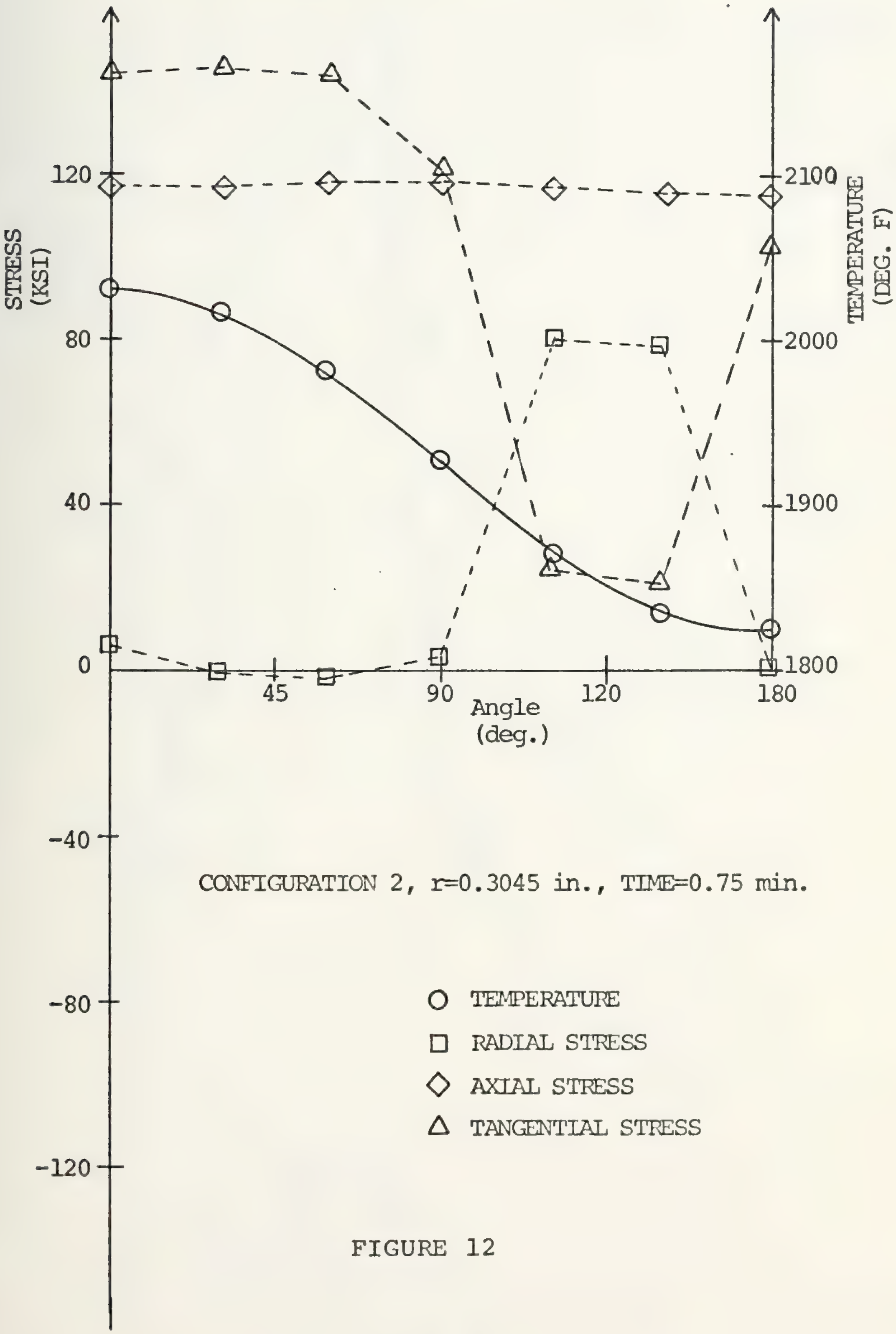


FIGURE 12

STRESS AND TEMPERATURE
VERSUS TIME

CONFIGURATION 2, r=0.3045 in., ANGLE=60°

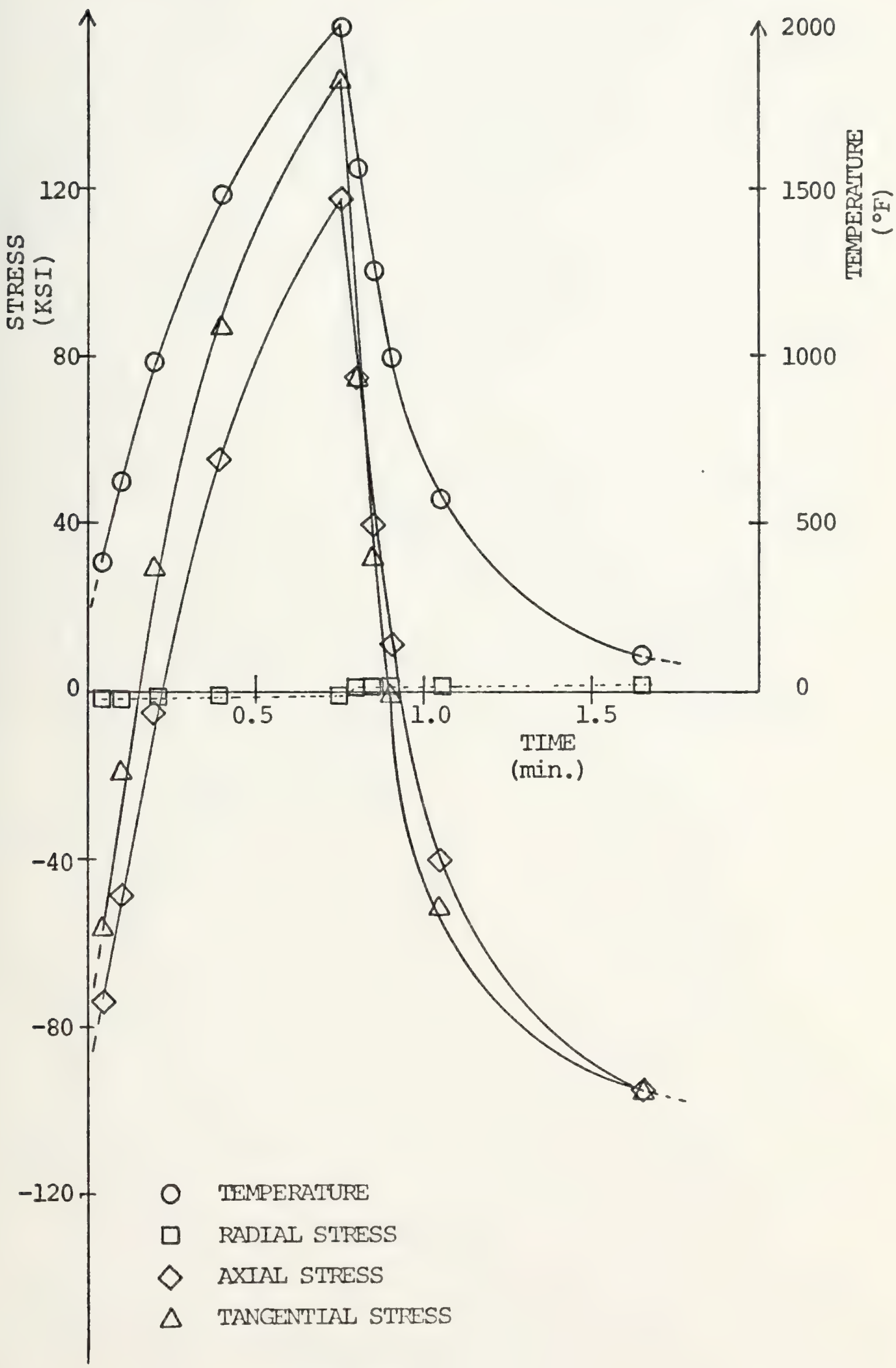


FIGURE 13

MAXIMUM AND MINIMUM PRINCIPAL STRESSES
VERSUS TIME
CONFIGURATION NUMBER TWO

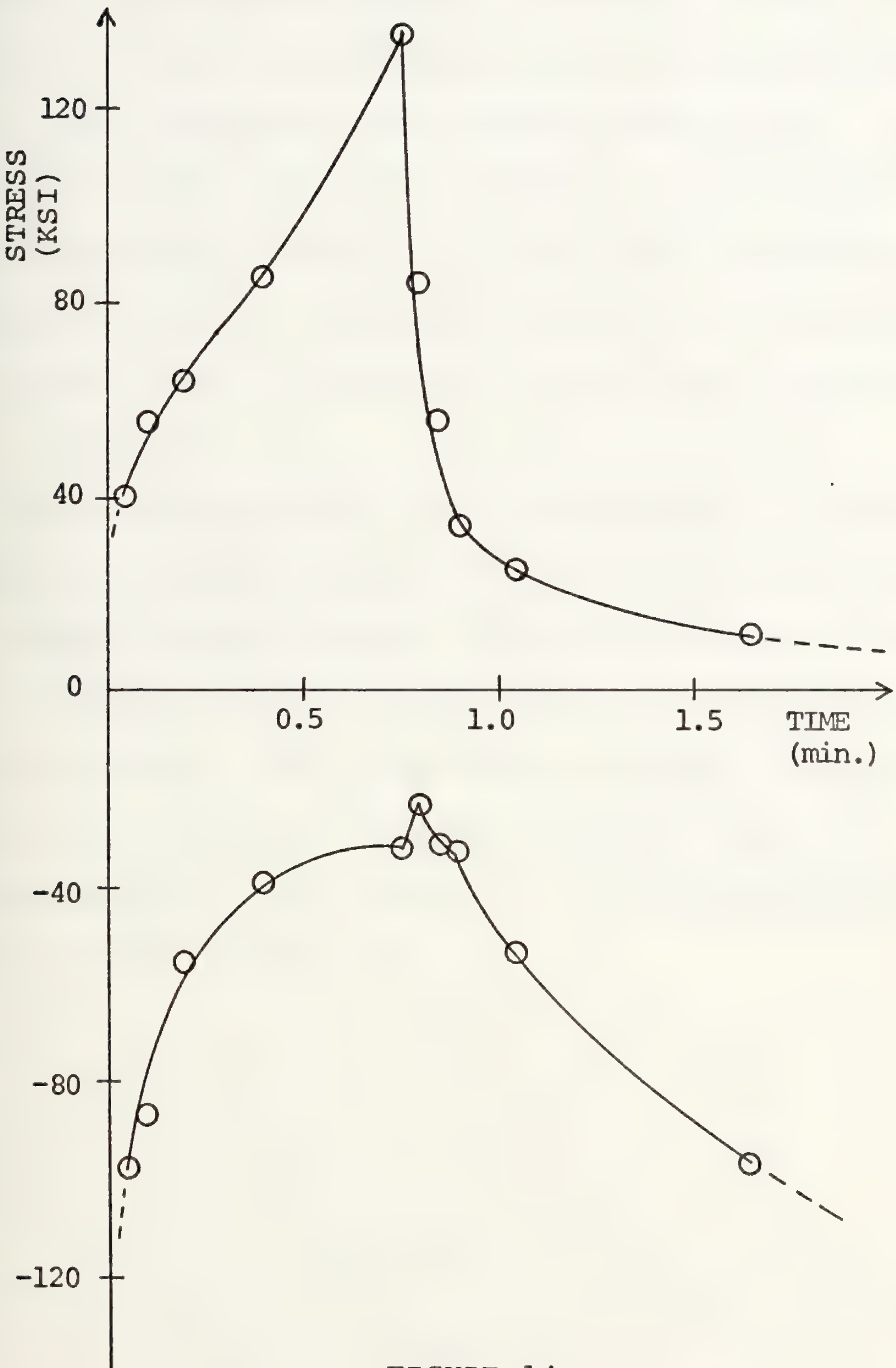


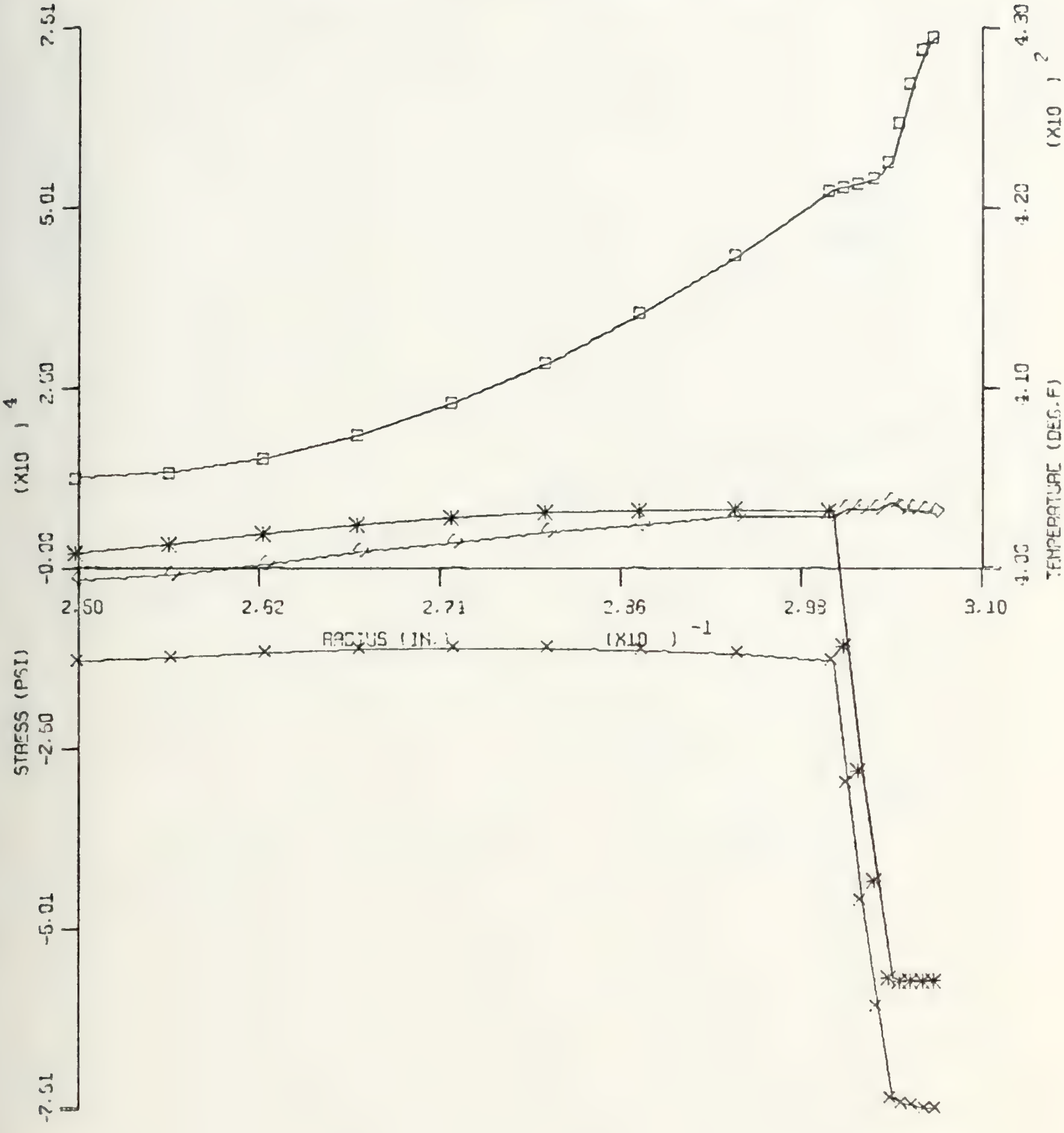
FIGURE 14

C. CONFIGURATION NUMBER THREE

This arrangement of coatings included a graded layer of nickel and zirconia. The effect of this layer was to eliminate the abrupt change in the stress levels seen in the first two coating arrangements. The maximum principal stress (134.9 KSI) occurred at the maximum temperature. The minimum principal stress (-97.2 KSI) occurred at the beginning and end of the thermal cycle, i.e., the room temperature condition. Since the tangential and axial stress levels are excessive for a major part of the cycle, failure would occur due to these stresses.

As with configuration two, a comparison of stress and temperature distributions as a function of angular position was prepared using data for the cylinder ends at a radius of 0.3055 inches and at a time of 0.75 minutes. The graphical representation of this comparison is shown in Figure 19. The function of stress and temperature versus time for the 60° angular position, at a radius of 0.3005 inches at the cylinder ends is shown in Figure 20.

STRESS AND TEMPERATURE
VERSUS RADIAL POSITION
TIME = 0.050 CONF.3-0



SIGMA R - DIAMOND SIGMA T - ASTERISK
SIGMA Z - CROSS TEMPERATURE - SQUARE

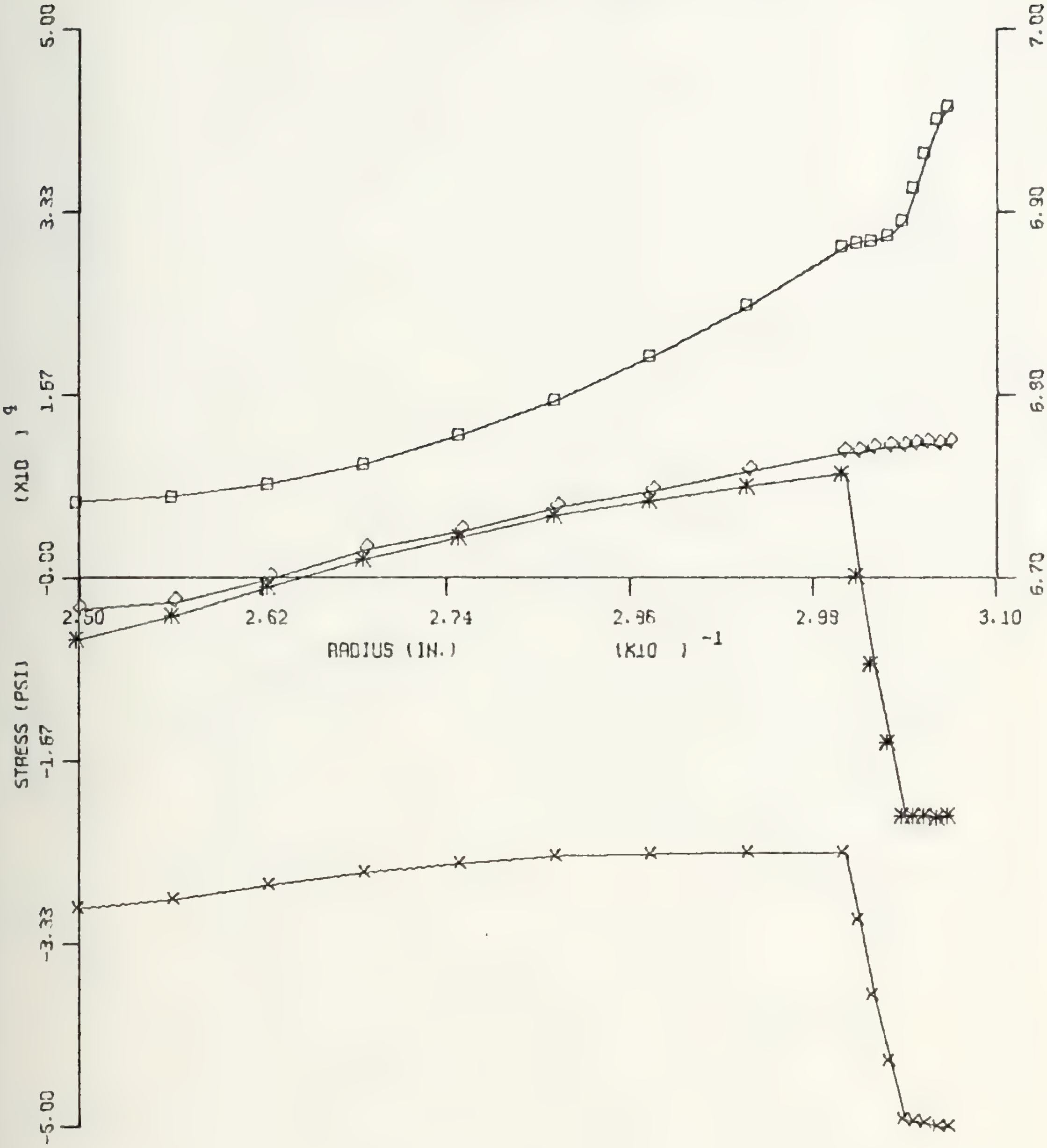
STRESS SCALE = 2.503×10^{4} PSI/INCH

TEMPERATURE SCALE = 10.0 DEGREES/INCH

RADIAL POSITION SCALE = 0.01 INCHES/INCH

FIGURE 15(a)

STRESS AND TEMPERATURE
VERSUS RADIAL POSITION
TIME = 0.100 CONF.3-0



SIGMA R - DIAMOND SIGMA T - ASTERISK
SIGMA Z - CROSS TEMPERATURE - SQUARE

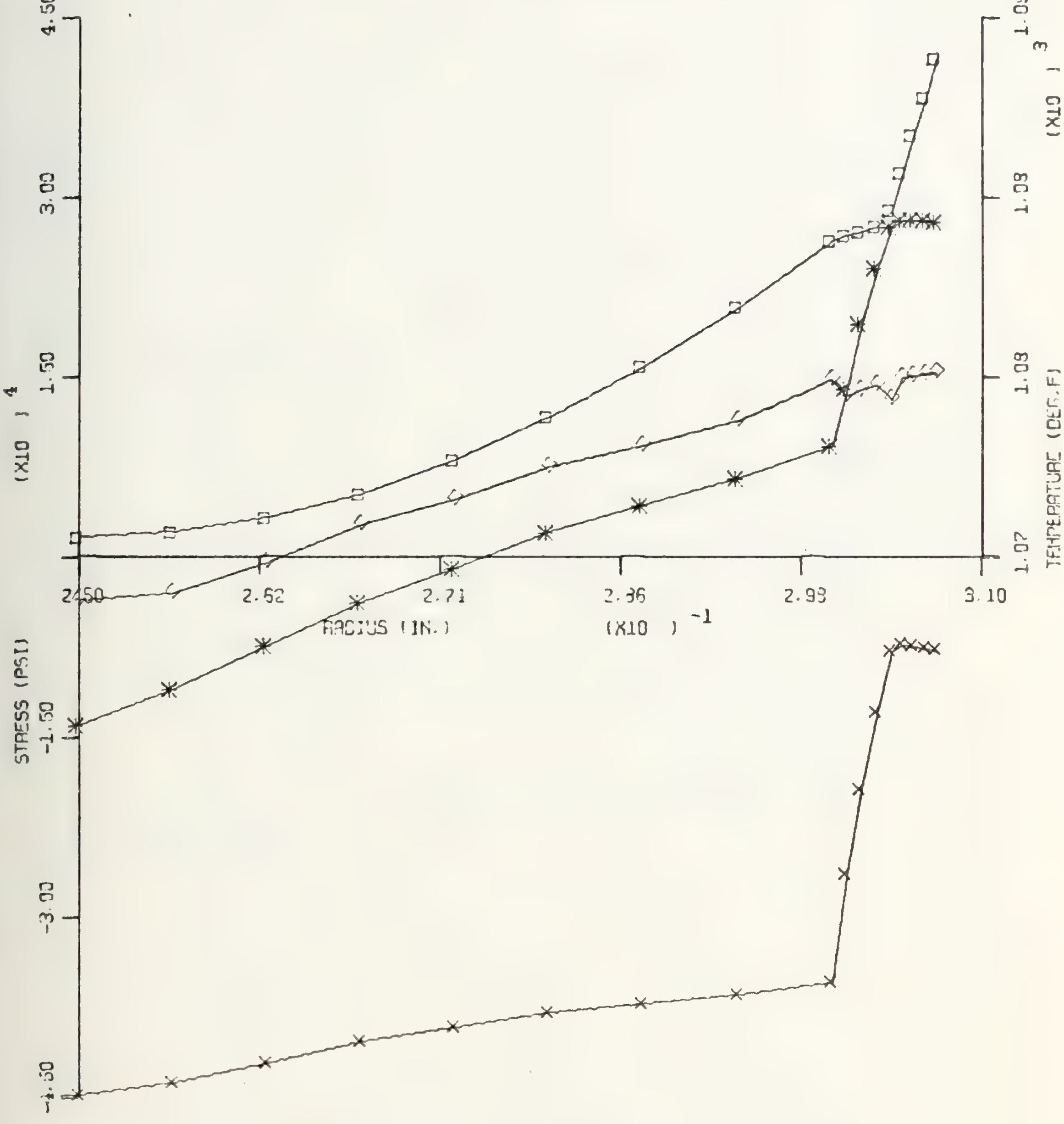
STRESS SCALE = 1.666X 10^4 PSI/INCH

TEMPERATURE SCALE = 10.0 DEGREES/INCH

RADIAL POSITION SCALE = 0.01 INCHES/INCH

FIGURE 15(b)

STRESS AND TEMPERATURE
VERSUS RADIAL POSITION
TIME = 0.200 CONF.3-0



SIGMA R - DIAMOND SIGMA T - ASTERISK
SIGMA Z - CROSS TEMPERATURE - SQUARE

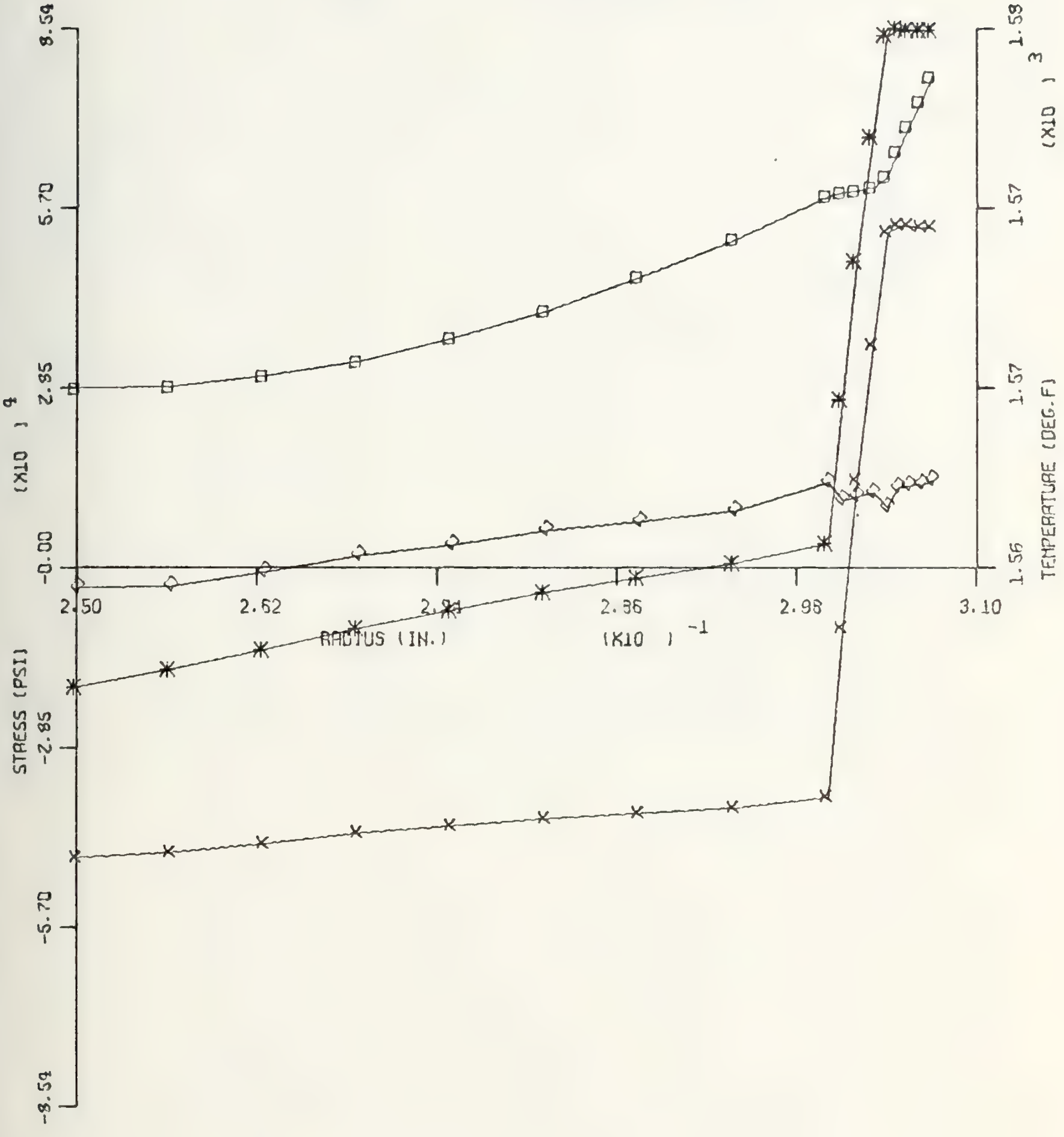
STRESS SCALE = 1.500X 10^3 PSI/INCH

TEMPERATURE SCALE = 6.7 DEGREES/INCH

RADIAL POSITION SCALE = 0.01 INCHES/INCH

FIGURE 15(c)

STRESS AND TEMPERATURE
VERSUS RADIAL POSITION
TIME = 0.400 CONF.3-0



SIGMA R - DIAMOND SIGMA T - ASTERISK
SIGMA Z - CROSS TEMPERATURE - SQUARE

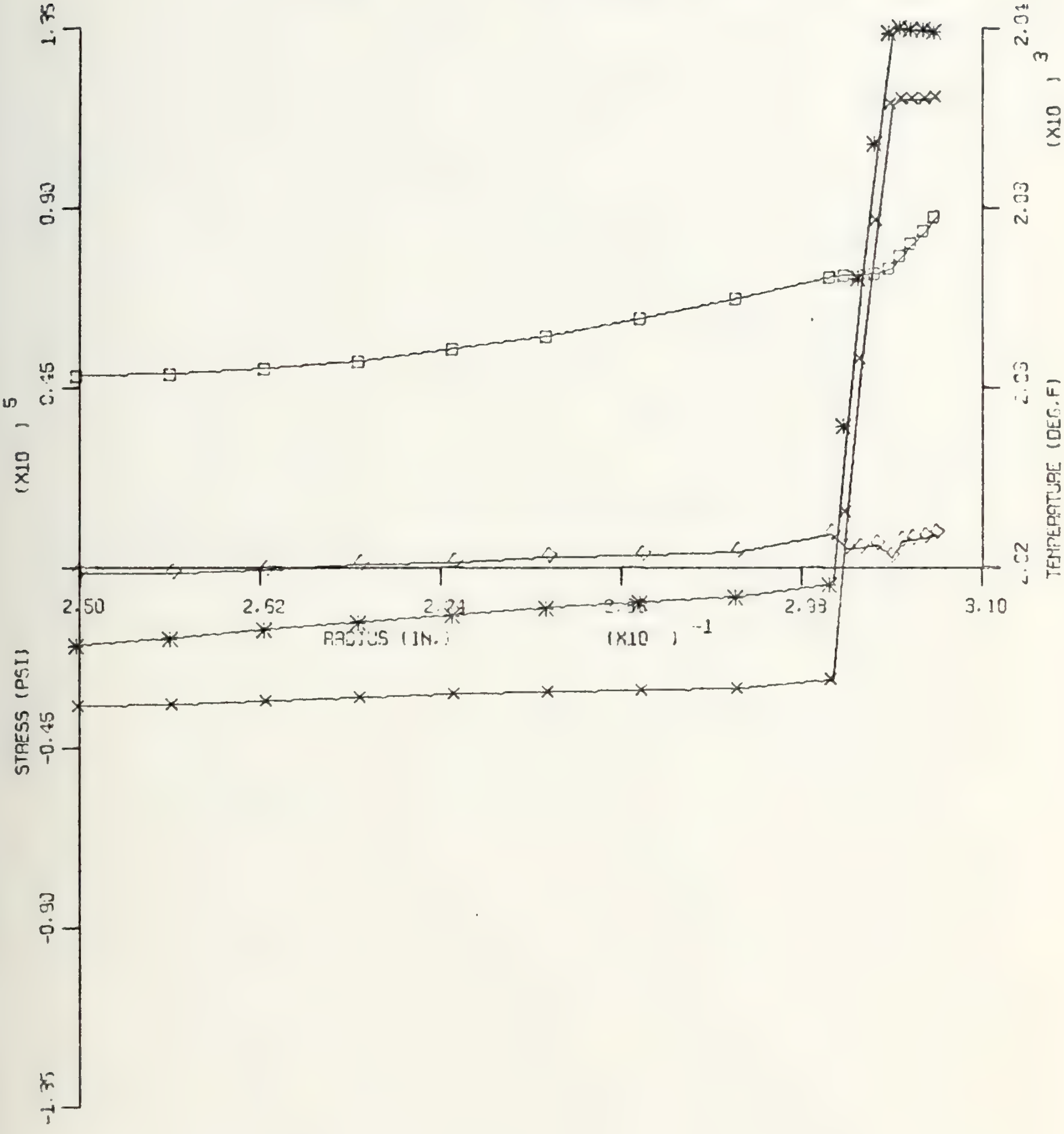
STRESS SCALE = 2.848X 10^{4} PSI/INCH

TEMPERATURE SCALE = 5.7 DEGREES/INCH

RADIAL POSITION SCALE = 0.01 INCHES/INCH

FIGURE 15(d)

STRESS AND TEMPERATURE
VERSUS RADIAL POSITION
TIME = 0.750 CONF.3-0



SIGMA R - DIAMOND SIGMA T - ASTERISK
SIGMA Z - CROSS TEMPERATURE - SQUARE

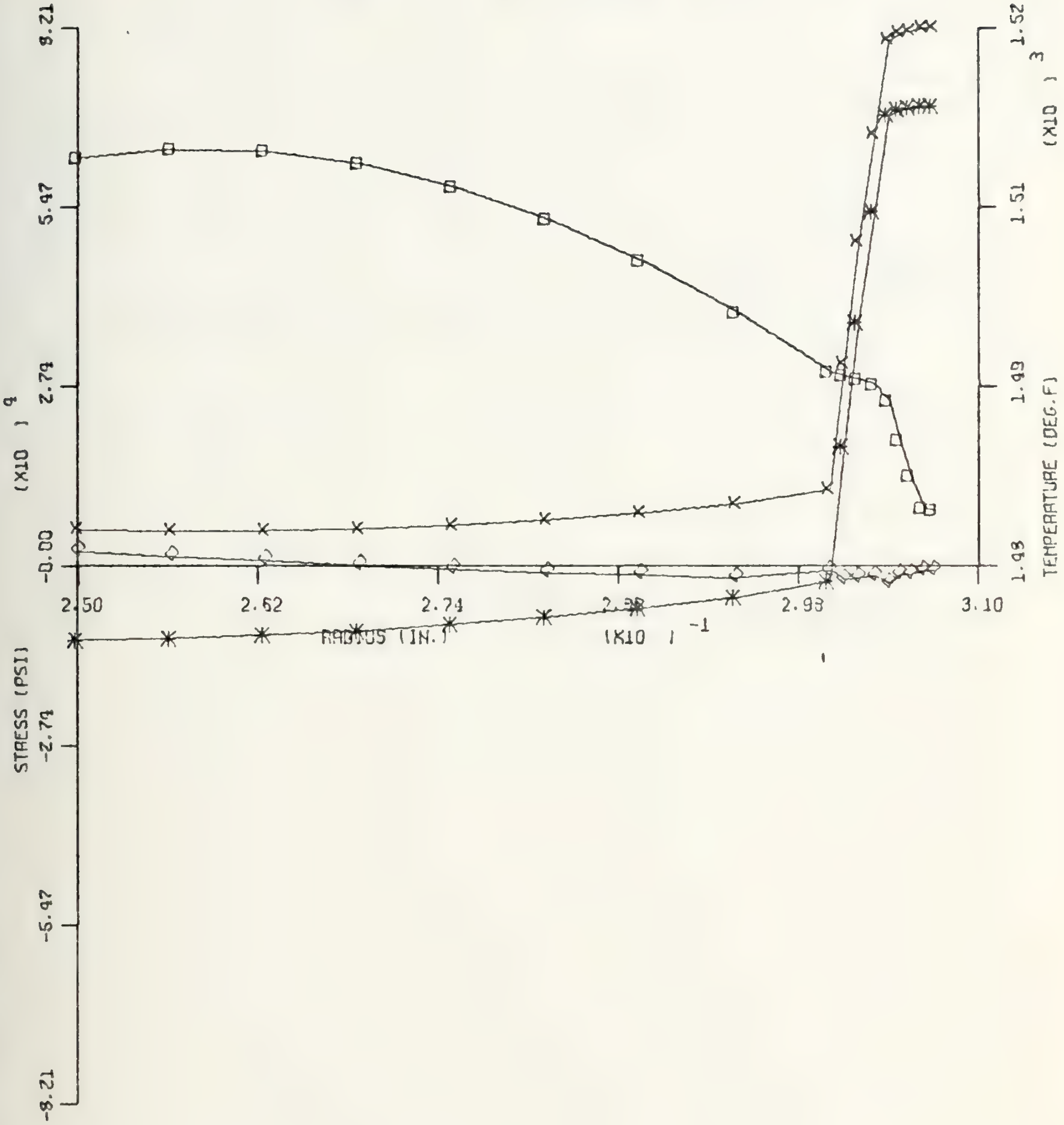
STRESS SCALE = 4.509X 10^4 PSI/INCH

TEMPERATURE SCALE = 6.7 DEGREES/INCH

RADIAL POSITION SCALE = 0.01 INCHES/INCH

FIGURE 15(e)

STRESS AND TEMPERATURE
VERSUS RADIAL POSITION
TIME = 0.800 CONF.3-0



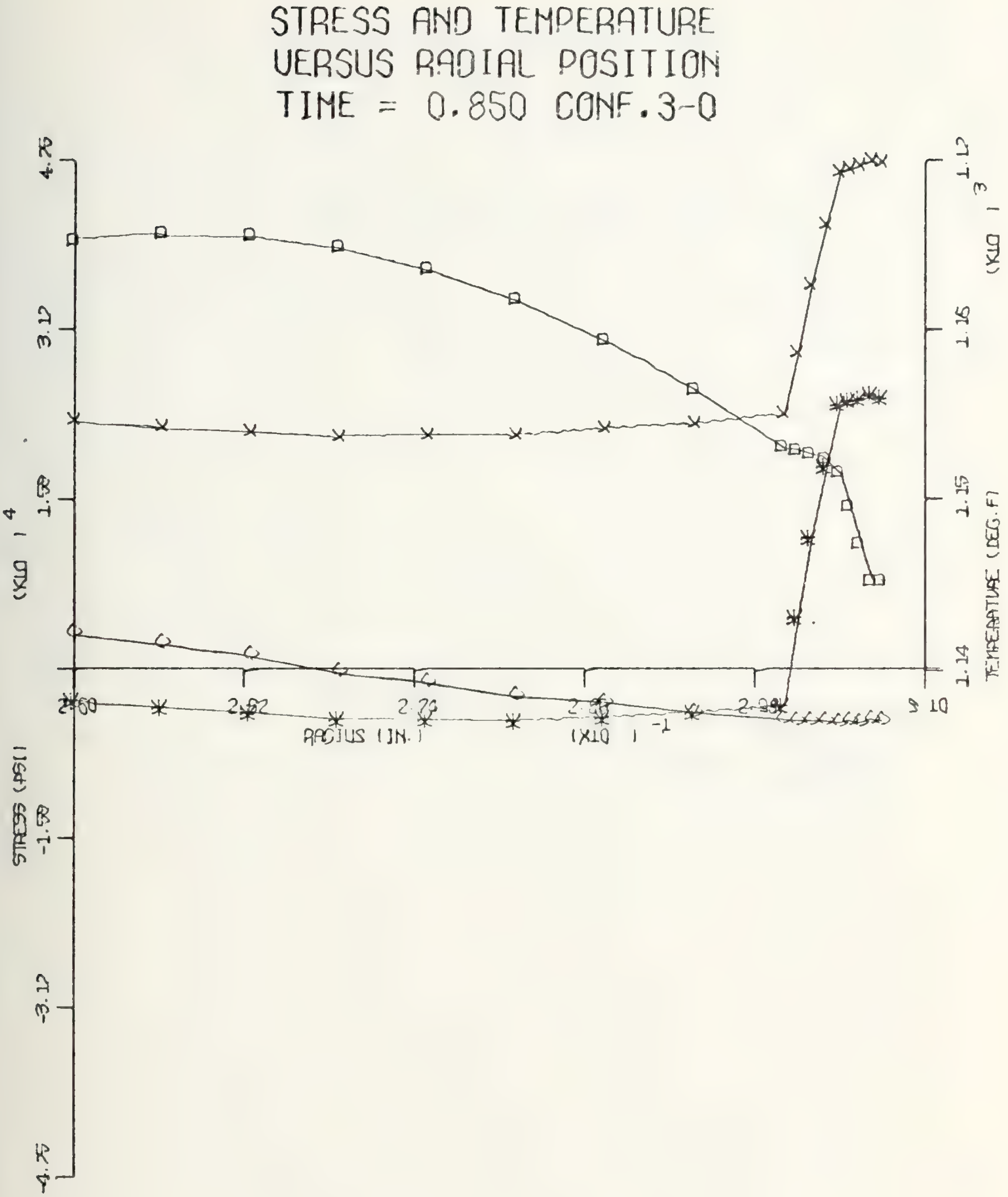
SIGMA R - DIAMOND SIGMA T - ASTERISK
SIGMA Z - CROSS TEMPERATURE - SQUARE

STRESS SCALE = 2.737K 10^4 PSI/INCH

TEMPERATURE SCALE = 13.3 DEGREES/INCH

RADIAL POSITION SCALE = 0.01 INCHES/INCH

FIGURE 15(f)



SIGMA R - DIAMOND SIGMA T - ASTERISK
SIGMA Z - CROSS TEMPERATURE - SQUARE

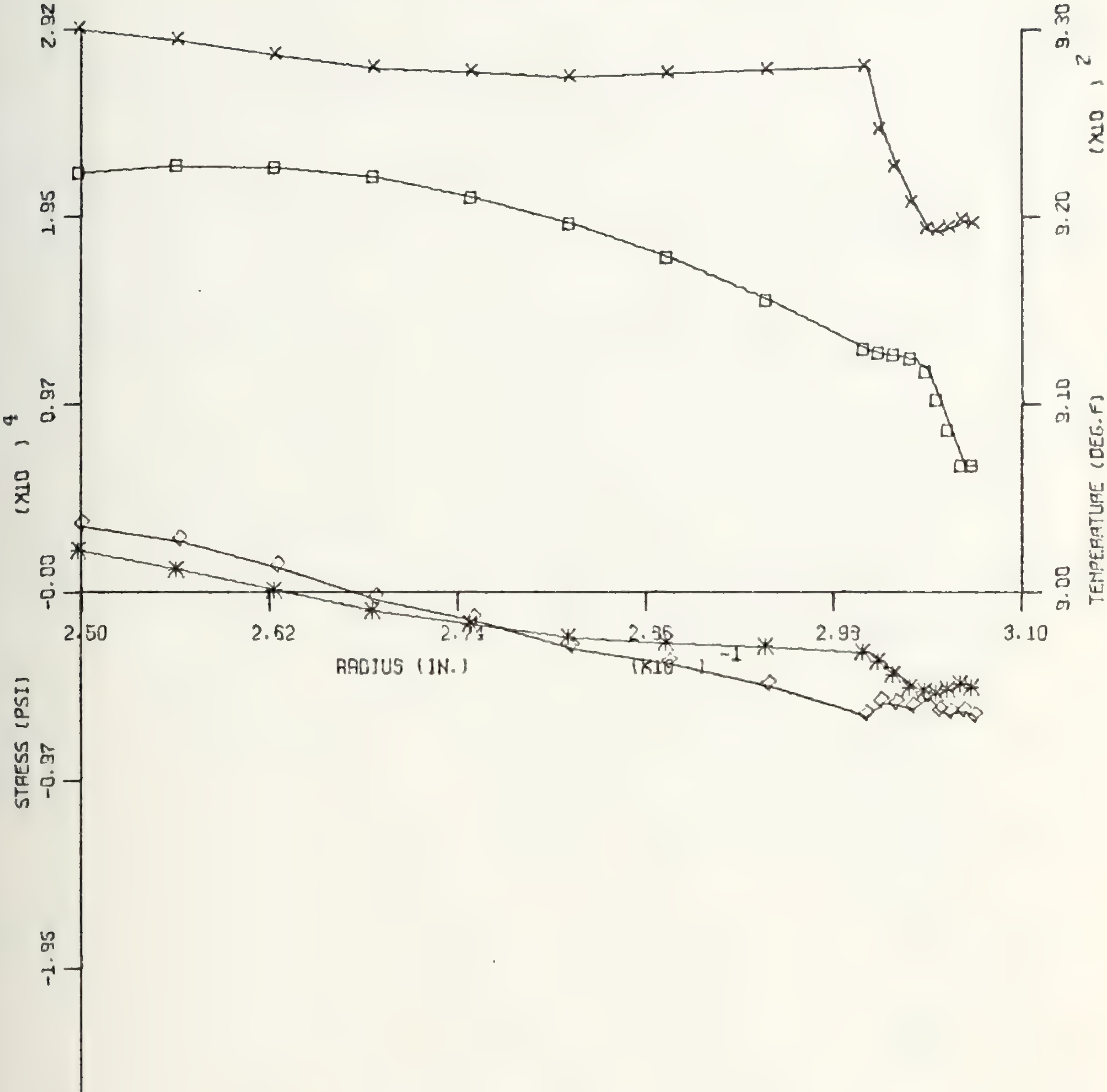
STRESS SCALE = 1.525X 10[4] PSI/INCH

TEMPERATURE SCALE = 10.0 DEGREES/INCH

RADIAL POSITION SCALE = 0.01 INCHES/INCH

FIGURE 15(g)

STRESS AND TEMPERATURE
VERSUS RADIAL POSITION
TIME = 0.900 CONF.3-0



SIGMA R - DIAMOND SIGMA T - ASTERISK
SIGMA Z - CROSS TEMPERATURE - SQUARE

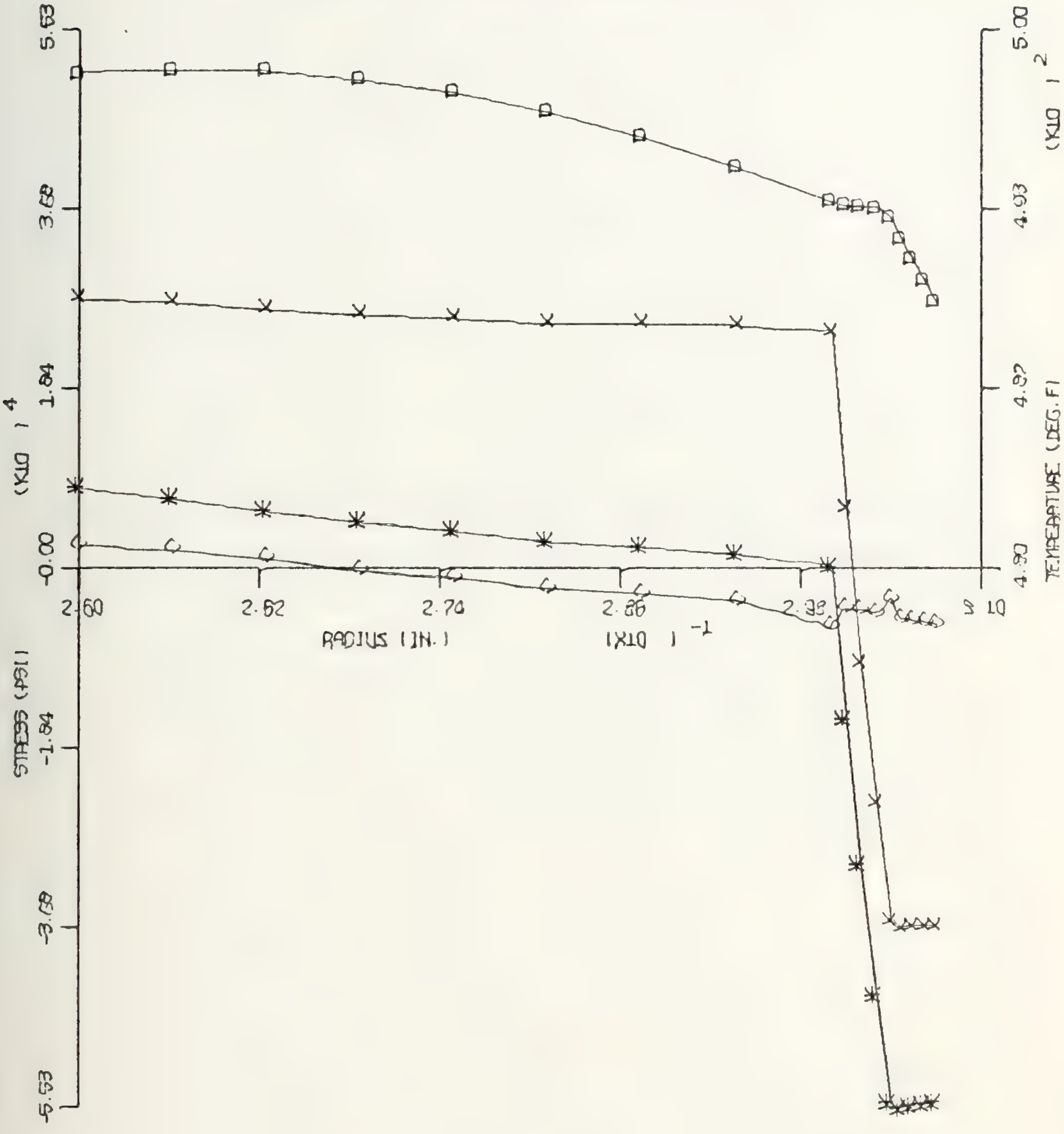
STRESS SCALE = 9.747X 10^{3} PSI/INCH

TEMPERATURE SCALE = 10.0 DEGREES/INCH

RADIAL POSITION SCALE = 0.01 INCHES/INCH

FIGURE 15(h)

STRESS AND TEMPERATURE
VERSUS RADIAL POSITION
TIME = 1.050 CONF.3-0



SIGMA R - DIAMOND SIGMA T - ASTERISK
SIGMA Z - CROSS TEMPERATURE - SQUARE

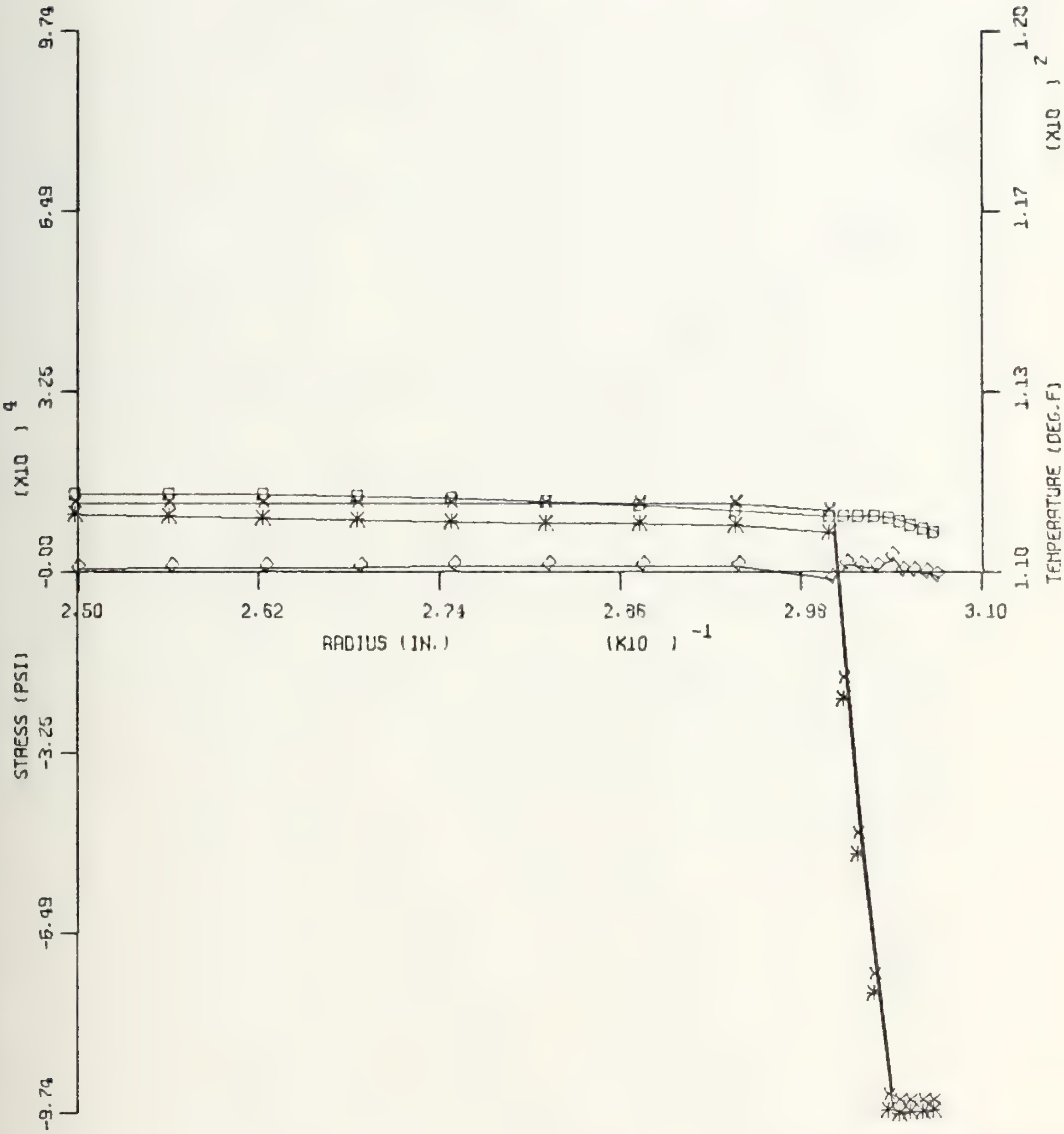
STRESS SCALE = 1.802X 10^{3} PSI/INCH

TEMPERATURE SCALE = 5.7 DEGREES/INCH

RADIAL POSITION SCALE = 0.01 INCHES/INCH

FIGURE 15(i)

STRESS AND TEMPERATURE
VERSUS RADIAL POSITION
TIME = 1.650 CONF.3-0



SIGMA R - DIAMOND SIGMA T - ASTERISK
SIGMA Z - CROSS TEMPERATURE - SQUARE

STRESS SCALE = 3.247X 10^4 PSI/INCH

TEMPERATURE SCALE = 3.3 DEGREES/INCH

RADIAL POSITION SCALE = 0.01 INCHES/INCH

FIGURE 15(j)

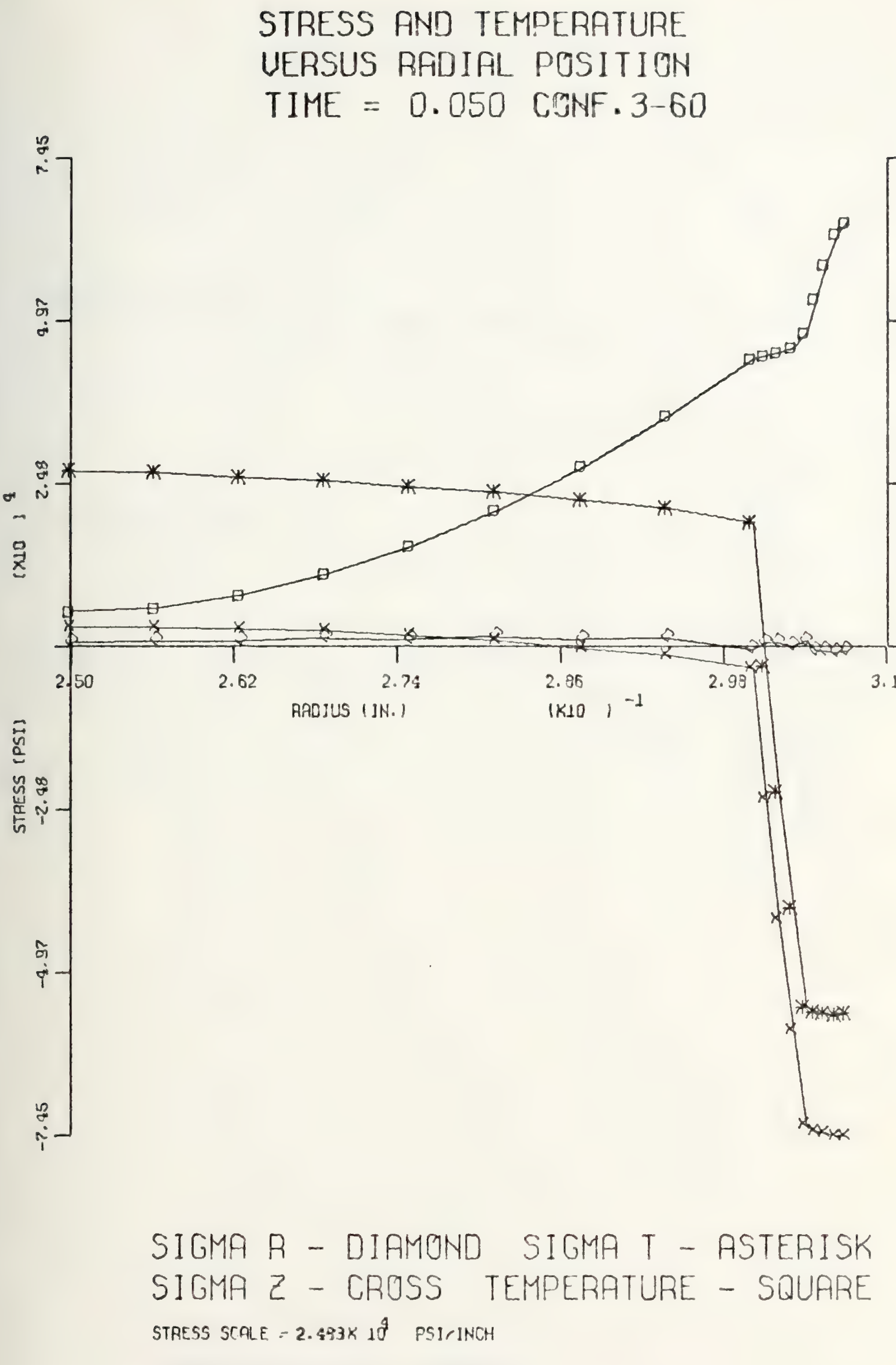
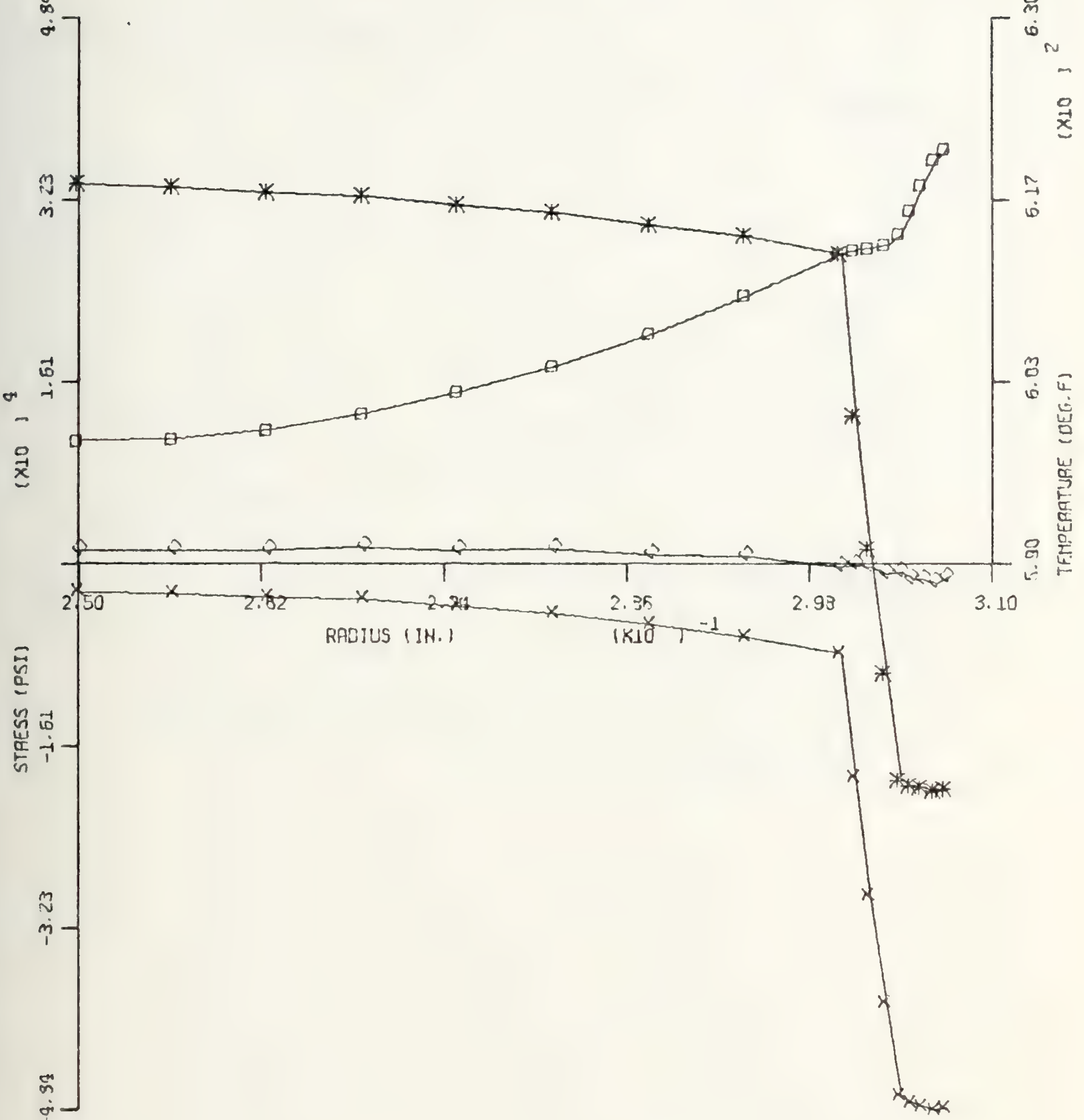

STRESS AND TEMPERATURE
VERSUS RADIAL POSITION
TIME = 0.050 CONF.3-60
7.45
4.97
2.48
-2.48
-4.97
-7.45
STRESS (PSI)
(X10) 4
3.90
3.80
3.70
3.60
TEMPERATURE (DEG.F)
(X10) 2
2.50
2.62
2.74
2.86
2.98
3.10
RADIUS (IN.)
(X10) -1
SIGMA R - DIAMOND SIGMA T - ASTERISK
SIGMA Z - CROSS TEMPERATURE - SQUARE
STRESS SCALE = 2.483X 10^4 PSI/INCH
TEMPERATURE SCALE = 10.0 DEGREES/INCH
RADIAL POSITION SCALE = 0.01 INCHES/INCH

FIGURE 16(a)

STRESS AND TEMPERATURE
VERSUS RADIAL POSITION
TIME = 0.100 CONF.3-60



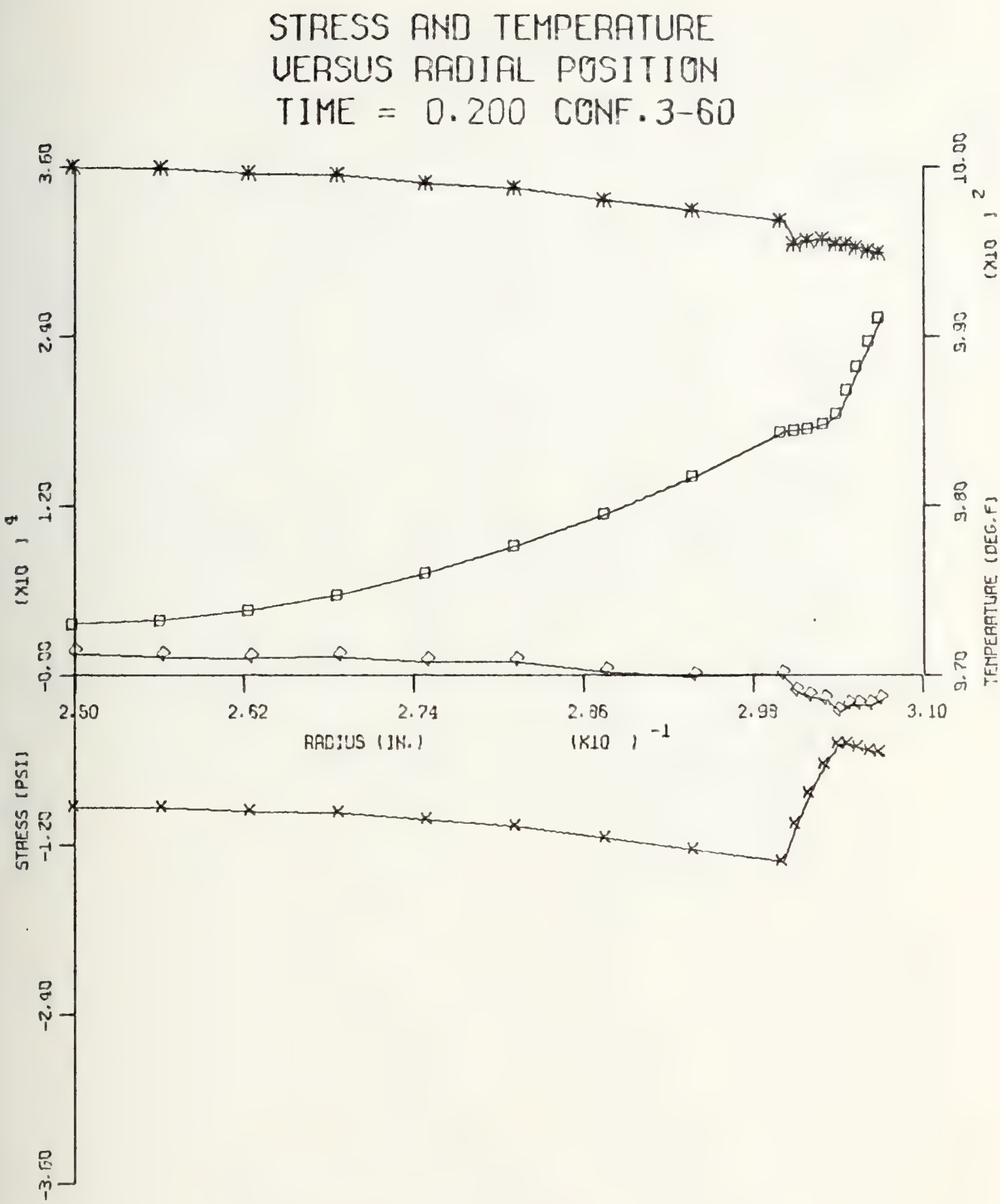
SIGMA R - DIAMOND SIGMA T - ASTERISK
SIGMA Z - CROSS TEMPERATURE - SQUARE

STRESS SCALE = 1.613X 10^{4} PSI/INCH

TEMPERATURE SCALE = 13.3 DEGREES/INCH

RADIAL POSITION SCALE = 0.01 INCHES/INCH

FIGURE 16(b)



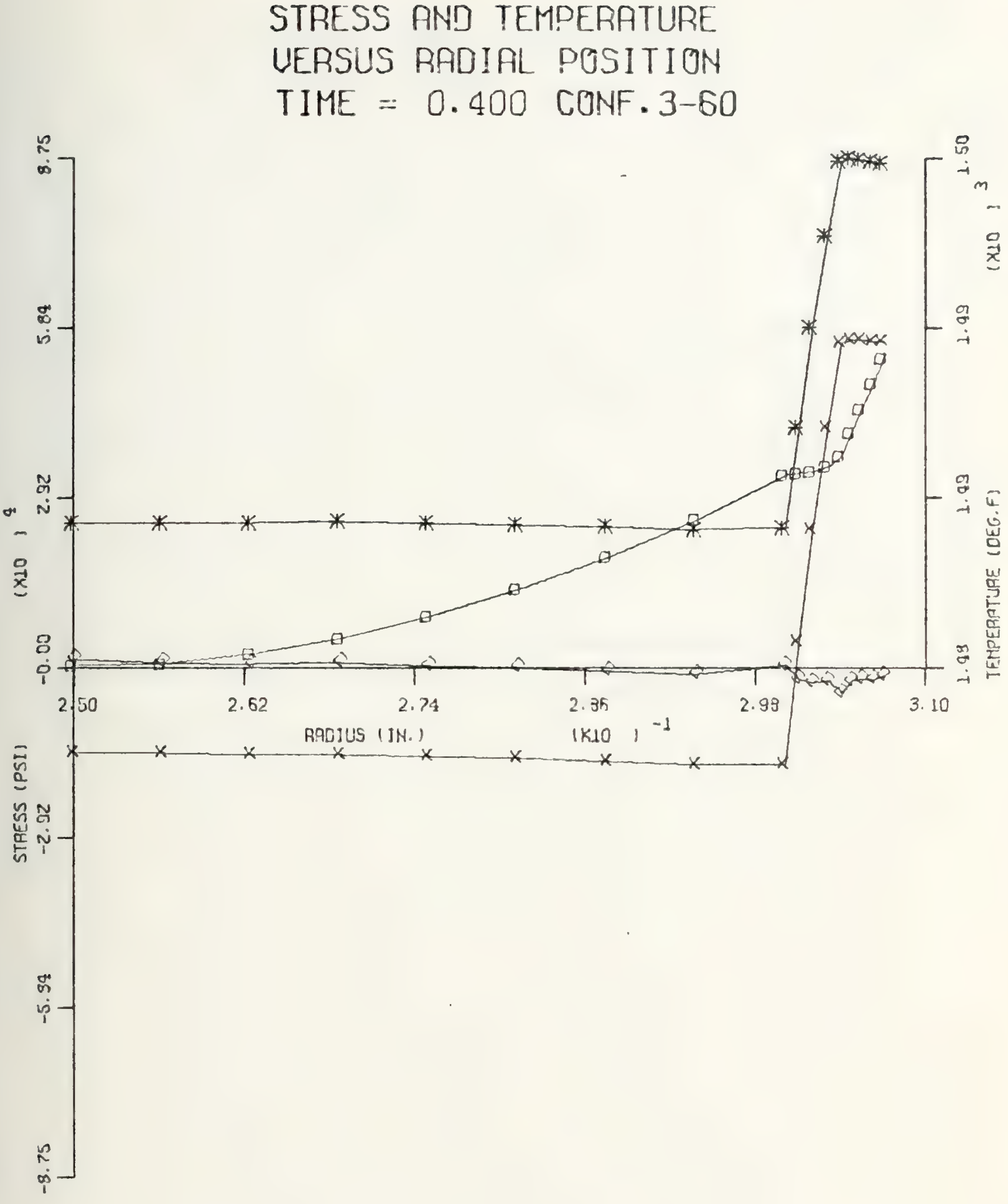
SIGMA R - DIAMOND SIGMA T - ASTERISK
SIGMA Z - CROSS TEMPERATURE - SQUARE

STRESS SCALE = 1.200X 10⁴ PSI/INCH

TEMPERATURE SCALE = 10.0 DEGREES/INCH

RADIAL POSITION SCALE = 0.01 INCHES/INCH

FIGURE 16(c)



SIGMA R - DIAMOND SIGMA T - ASTERISK
SIGMA Z - CROSS TEMPERATURE - SQUARE

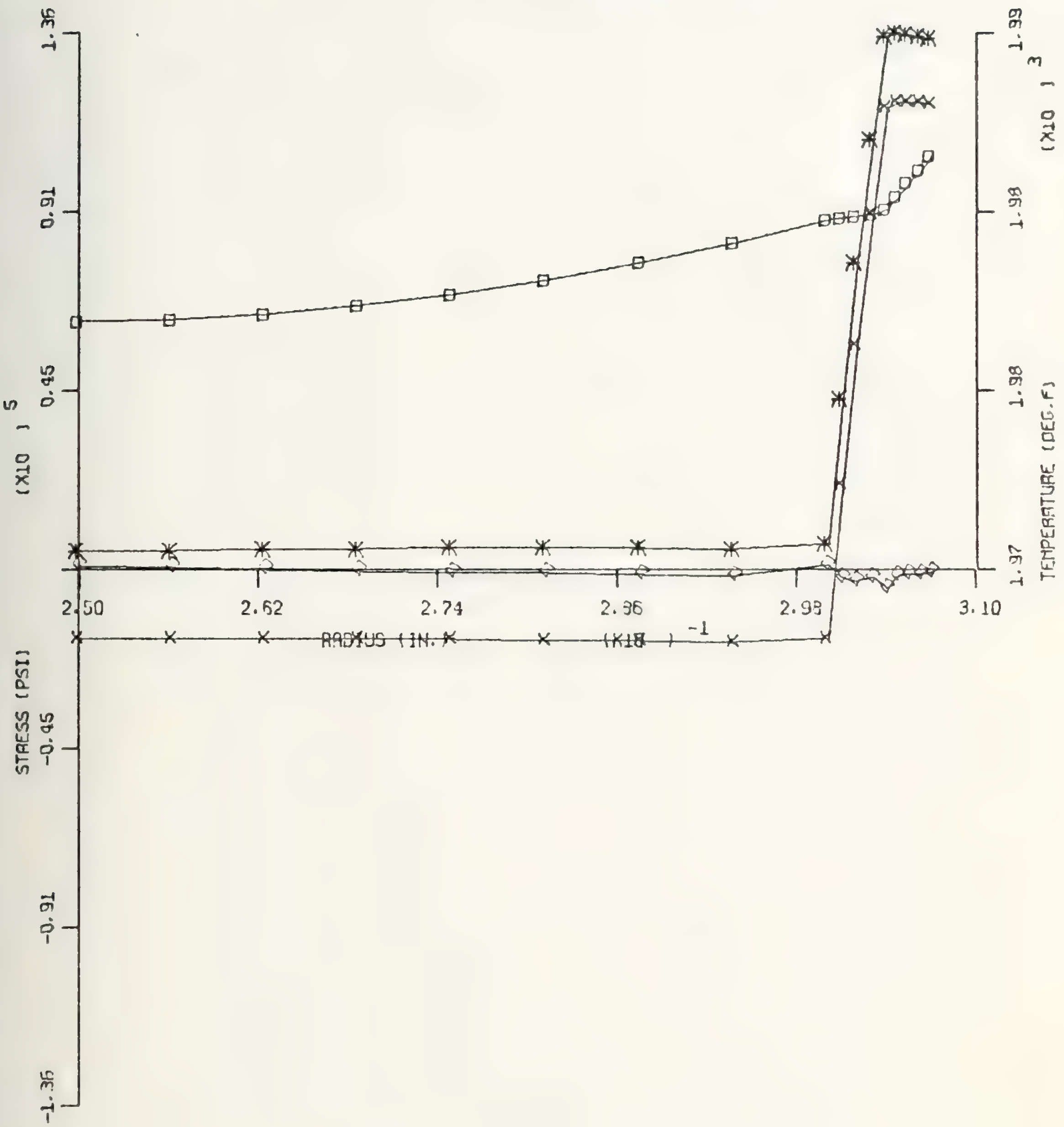
STRESS SCALE = 2.918X 10^4 PSI/INCH

TEMPERATURE SCALE = 6.7 DEGREES/INCH

RADIAL POSITION SCALE = 0.01 INCHES/INCH

FIGURE 16(d)

STRESS AND TEMPERATURE
VERSUS RADIAL POSITION
TIME = 0.750 CONF.3-60



SIGMA R - DIAMOND SIGMA T - ASTERISK
SIGMA Z - CROSS TEMPERATURE - SQUARE

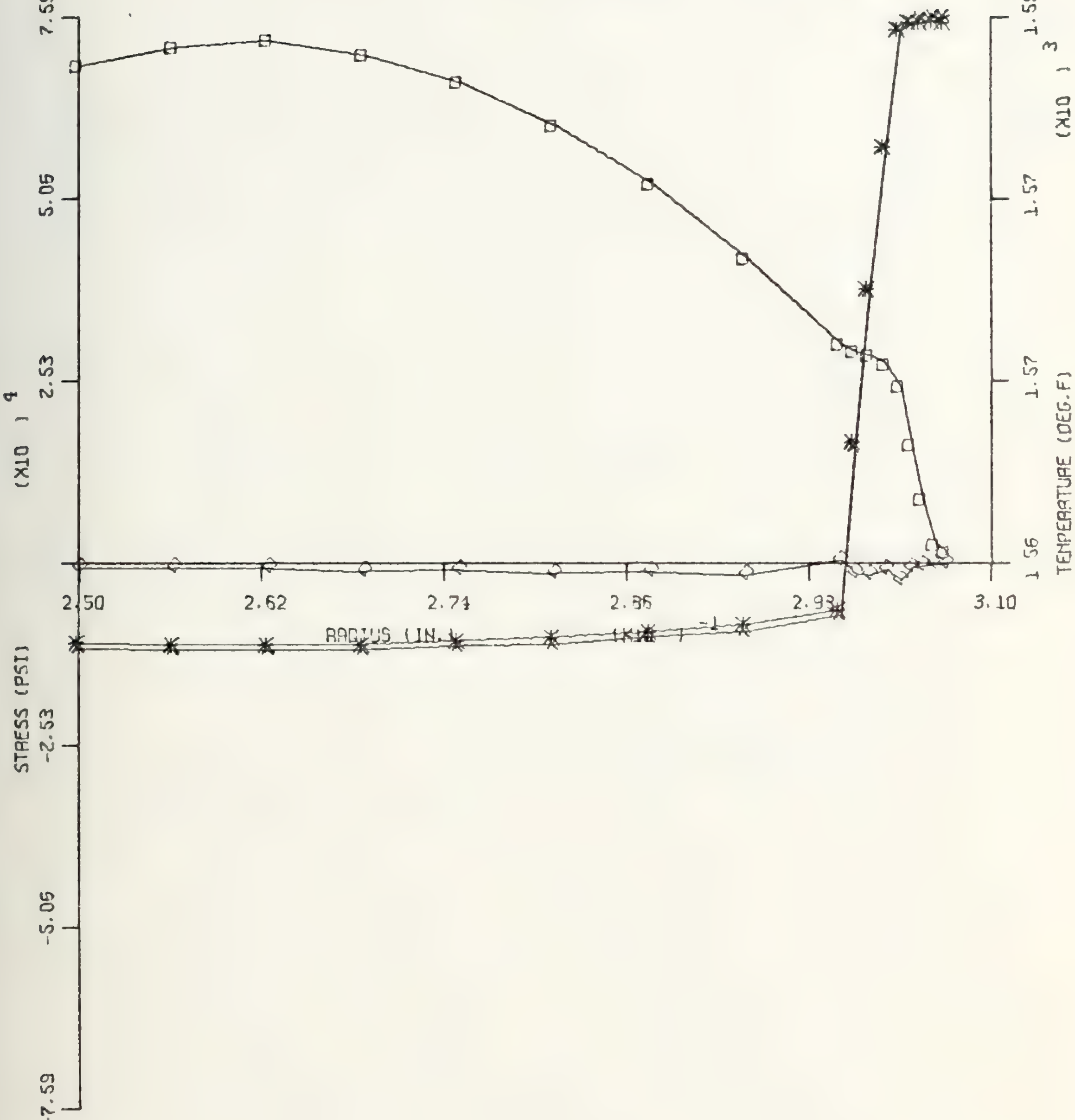
STRESS SCALE = 4.547X 10^4 PSI/INCH

TEMPERATURE SCALE = 6.7 DEGREES/INCH

RADIAL POSITION SCALE = 0.01 INCHES/INCH

FIGURE 16(e)

STRESS AND TEMPERATURE
VERSUS RADIAL POSITION
TIME = 0.800 CONF.3-60



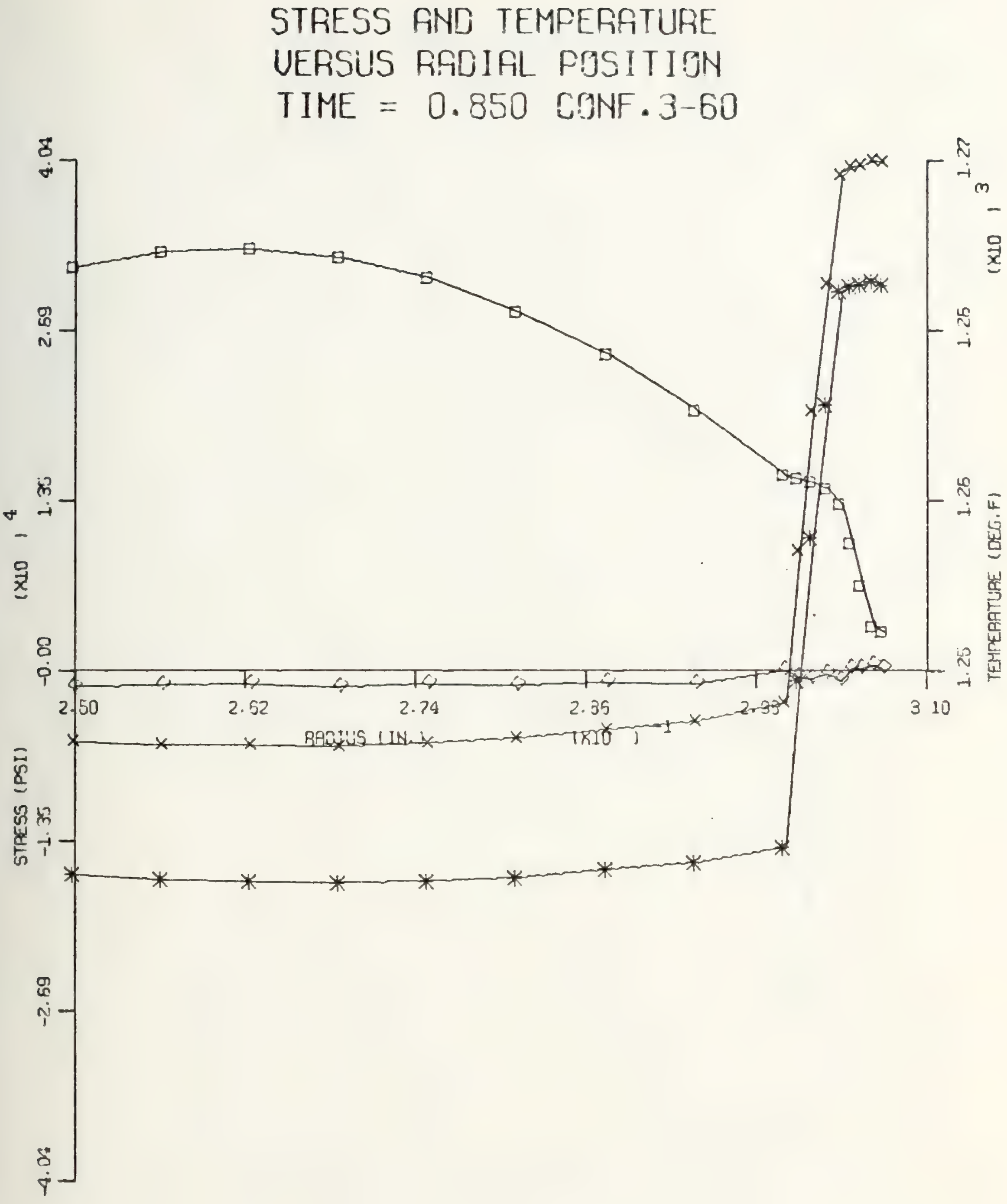
SIGMA R - DIAMOND SIGMA T - ASTERISK
SIGMA Z - CROSS TEMPERATURE - SQUARE

STRESS SCALE = 2.529X 10^4 PSI/INCH

TEMPERATURE SCALE = 6.7 DEGREES/INCH

RADIAL POSITION SCALE = 0.01 INCHES/INCH

FIGURE 16(f)



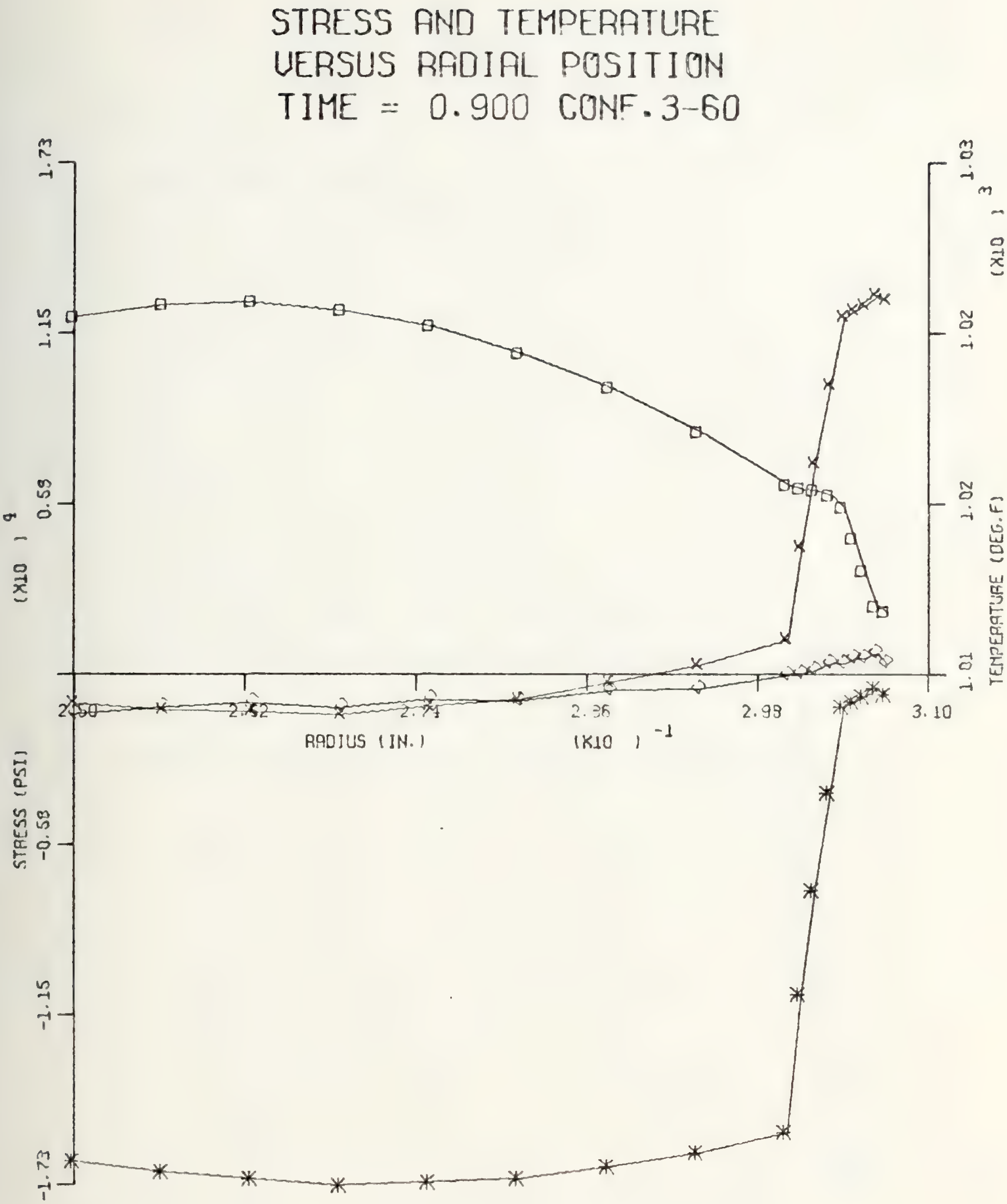
SIGMA R - DIAMOND SIGMA T - ASTERISK
SIGMA Z - CROSS TEMPERATURE - SQUARE

STRESS SCALE = 1.345 X 10^4 PSI/INCH

TEMPERATURE SCALE = 6.7 DEGREES/INCH

RADIAL POSITION SCALE = 0.01 INCHES/INCH

FIGURE 16(g)



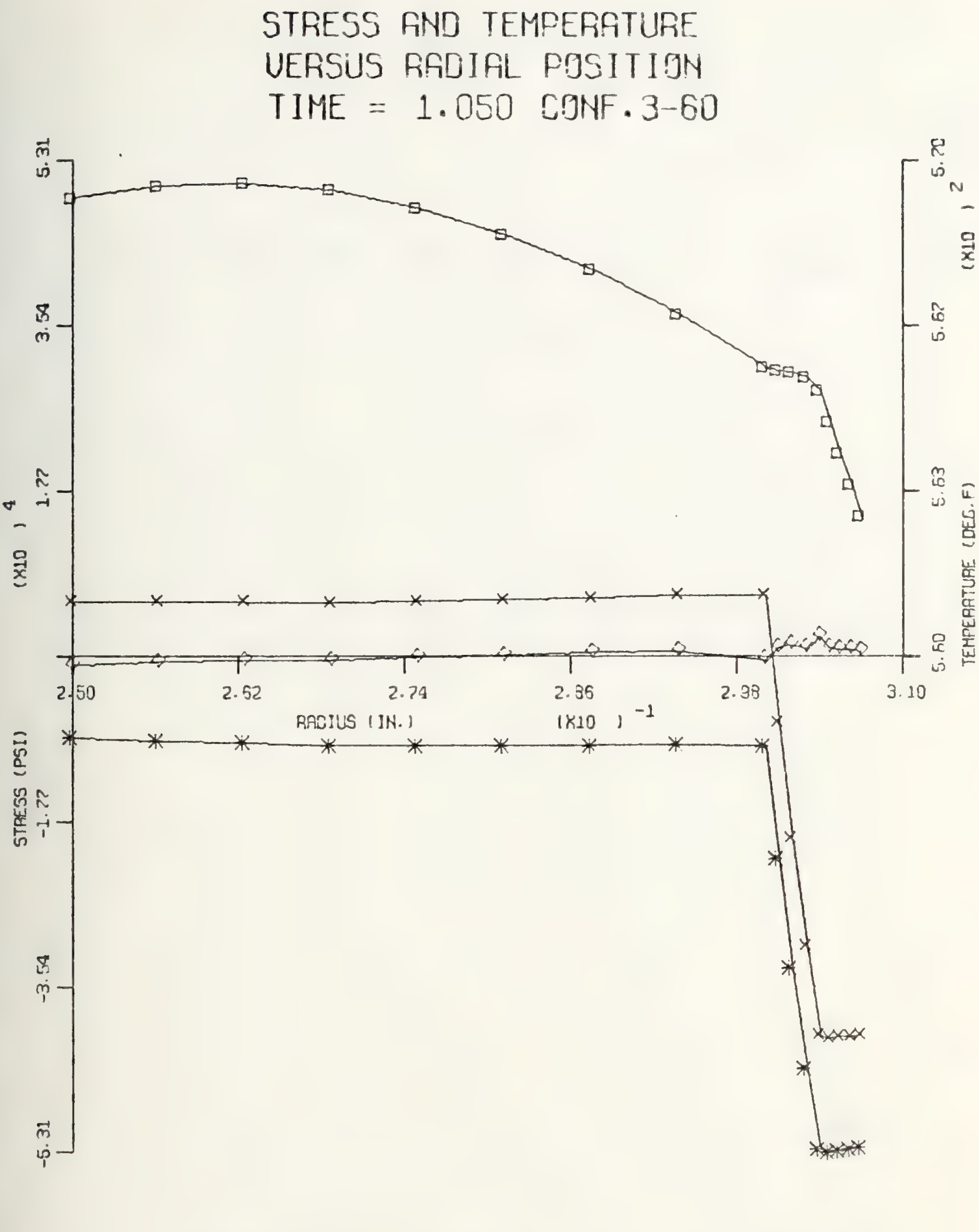
SIGMA R - DIAMOND SIGMA T - ASTERISK
SIGMA Z - CROSS TEMPERATURE - SQUARE

STRESS SCALE = 5.772K 10^3 PSI/INCH

TEMPERATURE SCALE = 6.7 DEGREES/INCH

RADIAL POSITION SCALE = 0.01 INCHES/INCH

FIGURE 16(h)



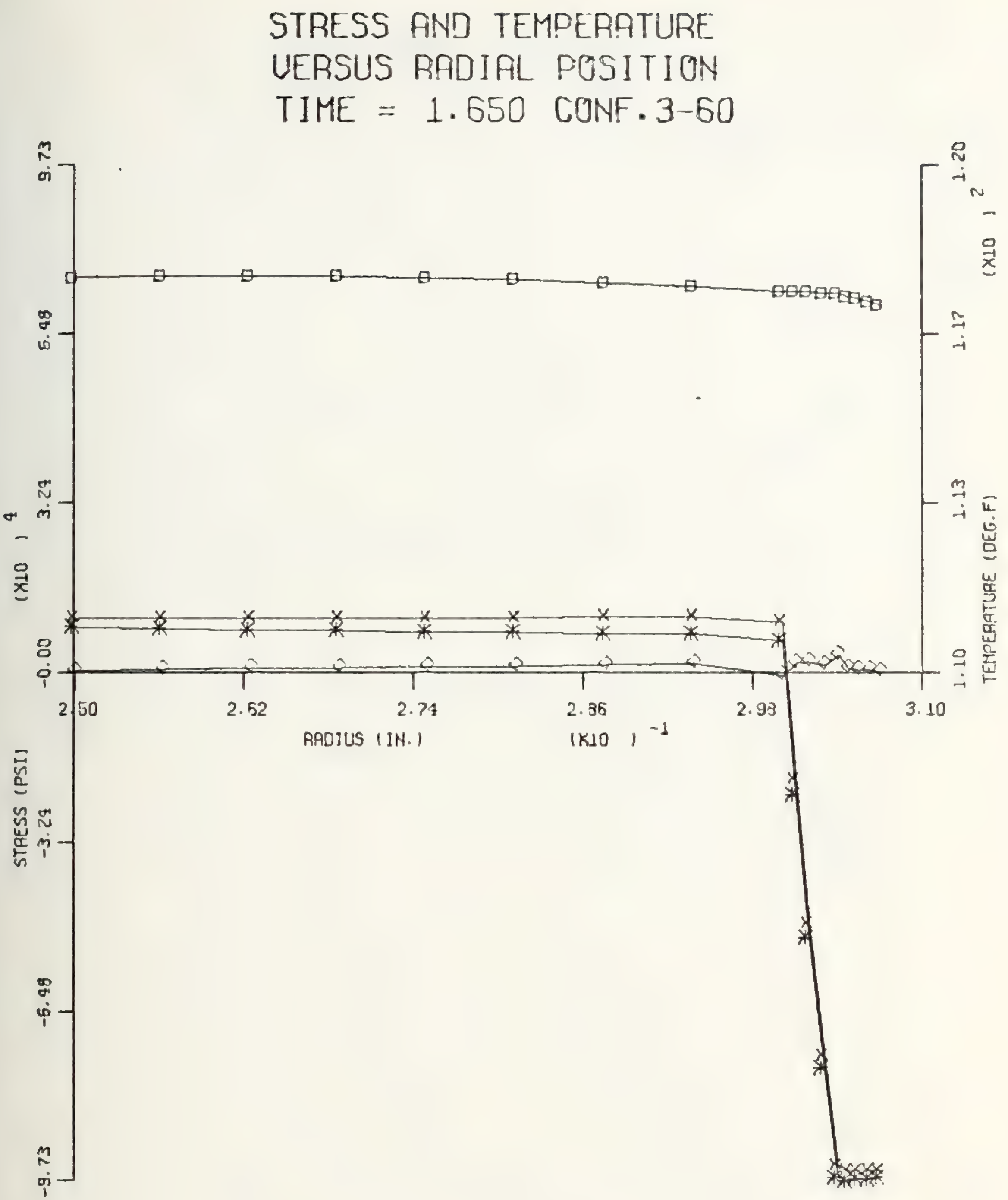
SIGMA R - DIAMOND SIGMA T - ASTERISK
SIGMA Z - CROSS TEMPERATURE - SQUARE

STRESS SCALE = 1.769X 10^4 PSI/INCH

TEMPERATURE SCALE = 3.3 DEGREES/INCH

RADIAL POSITION SCALE = 0.01 INCHES/INCH

FIGURE 16(i)



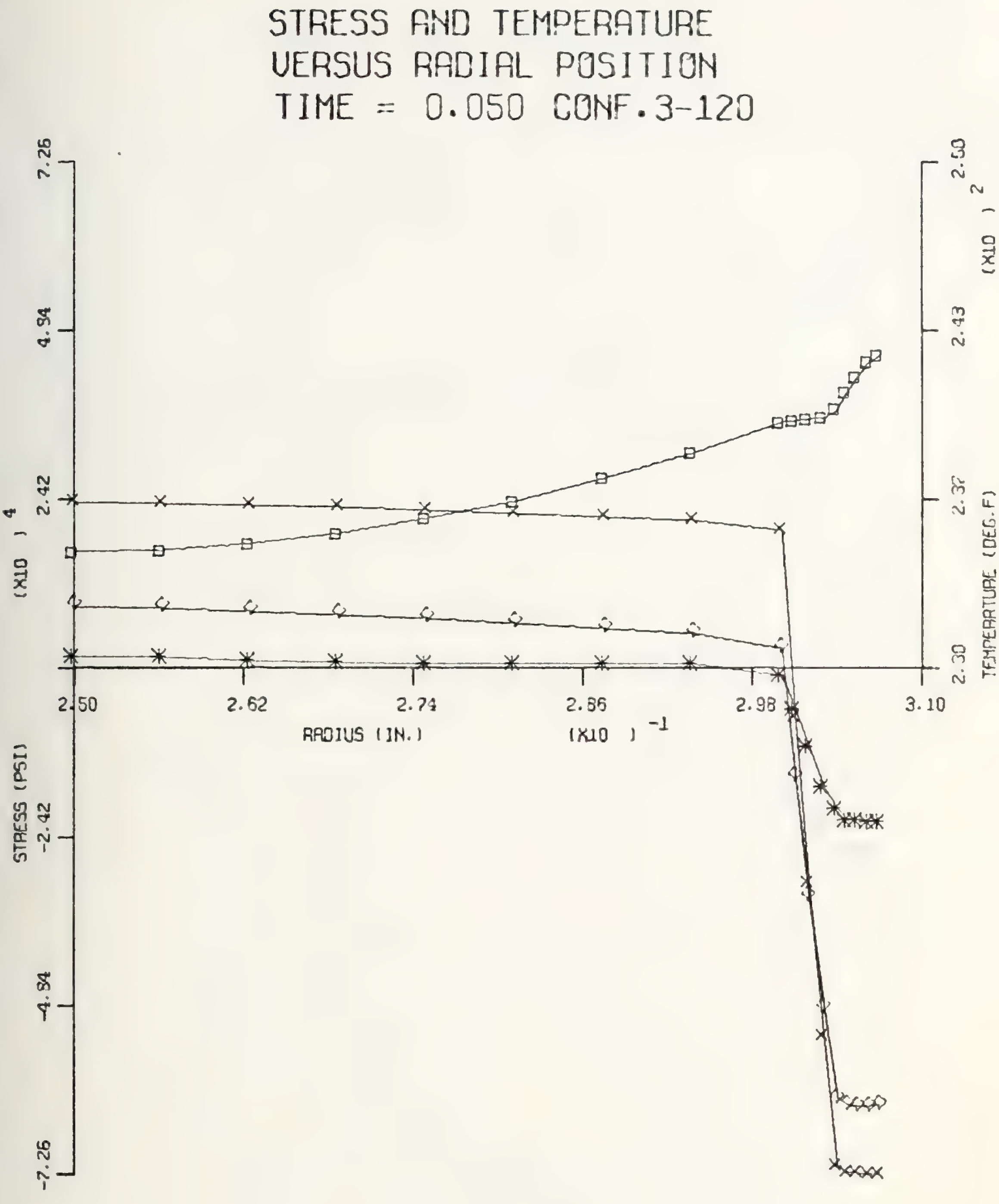
SIGMA R - DIAMOND SIGMA T - ASTERISK
SIGMA Z - CROSS TEMPERATURE - SQUARE

STRESS SCALE = 3.242X 10^4 PSI/INCH

TEMPERATURE SCALE = 3.3 DEGREES/INCH

RADIAL POSITION SCALE = 0.01 INCHES/INCH

FIGURE 16(j)



SIGMA R - DIAMOND SIGMA T - ASTERISK
SIGMA Z - CROSS TEMPERATURE - SQUARE

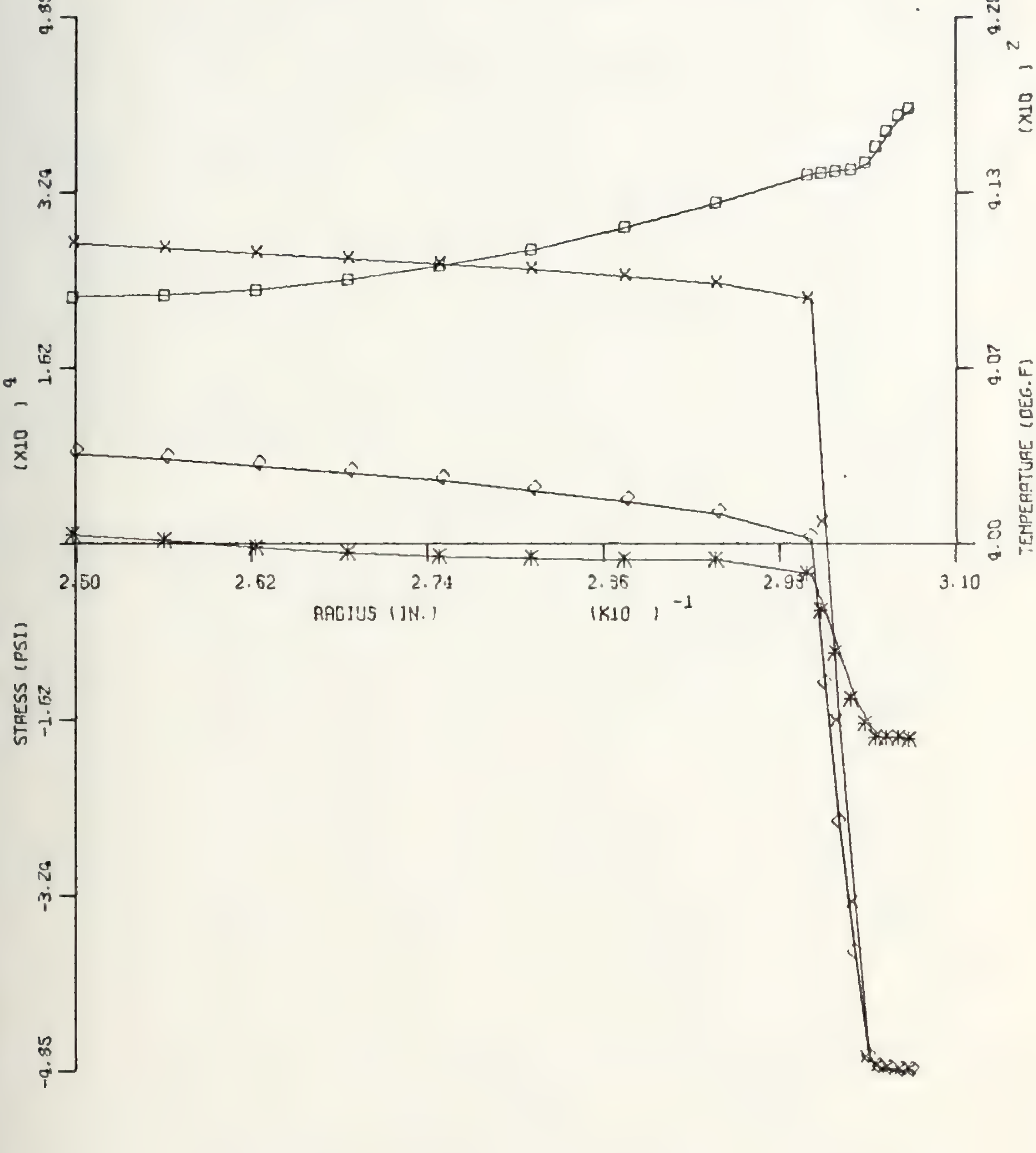
STRESS SCALE = 2.421 X 10^4 PSI/INCH

TEMPERATURE SCALE = 6.7 DEGREES/INCH

RADIAL POSITION SCALE = 0.01 INCHES/INCH

FIGURE 17(a)

STRESS AND TEMPERATURE
VERSUS RADIAL POSITION
TIME = 0.100 CONF.3-120



SIGMA R - DIAMOND SIGMA T - ASTERISK
SIGMA Z - CROSS TEMPERATURE - SQUARE

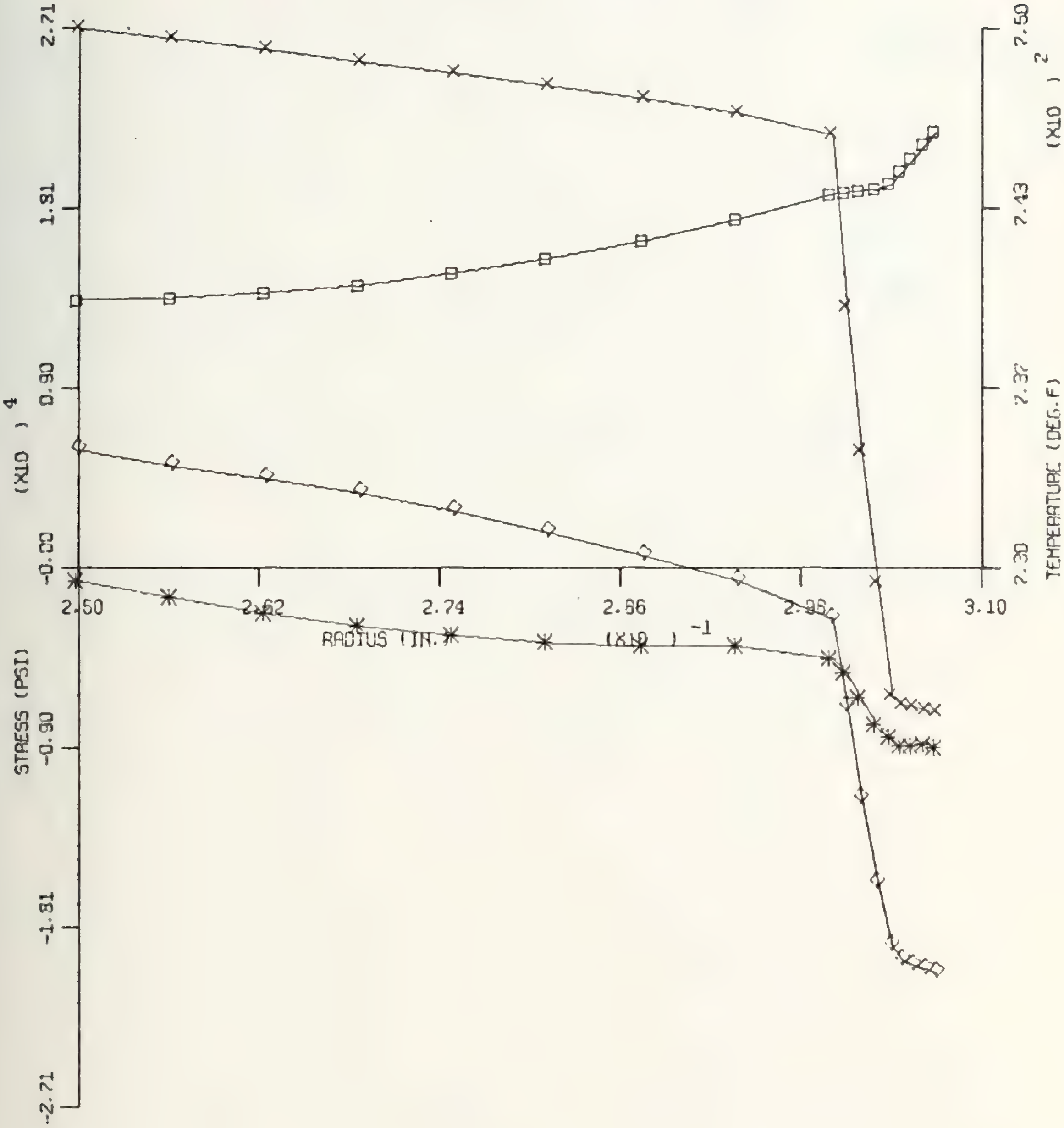
STRESS SCALE = 1.518K 10^{4} PSI/INCH

TEMPERATURE SCALE = 6.7 DEGREES/INCH

RADIAL POSITION SCALE = 0.01 INCHES/INCH

FIGURE 17(b)

STRESS AND TEMPERATURE
VERSUS RADIAL POSITION
TIME = 0.200 CONF.3-120



SIGMA R - DIAMOND SIGMA T - ASTERISK
SIGMA Z - CROSS TEMPERATURE - SQUARE

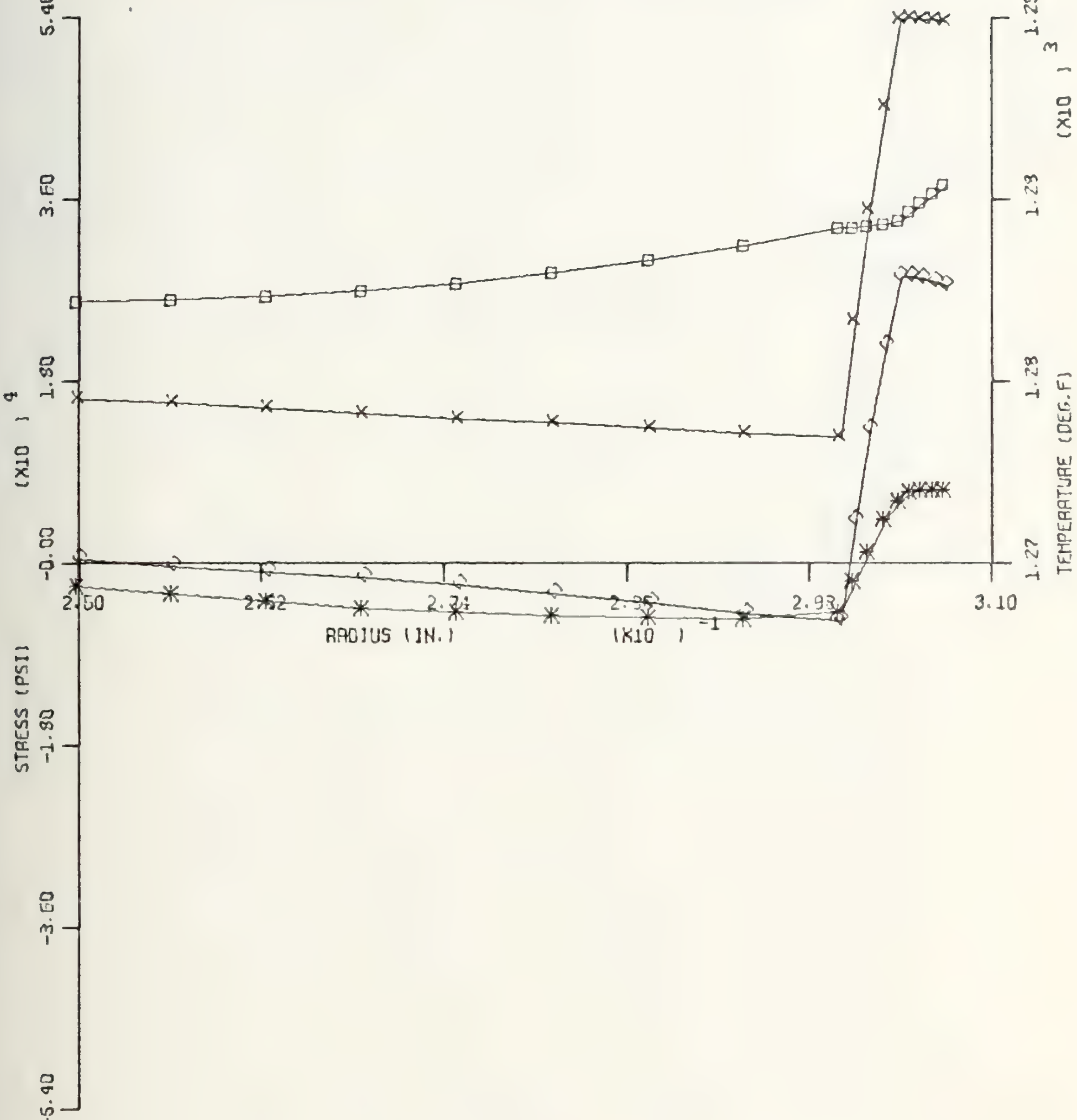
STRESS SCALE = 9.033X 10^3 PSI/INCH

TEMPERATURE SCALE = 6.7 DEGREES/INCH

RADIAL POSITION SCALE = 0.01 INCHES/INCH

FIGURE 17(c)

STRESS AND TEMPERATURE
VERSUS RADIAL POSITION
TIME = 0.400 CONF.3-120



SIGMA R - DIAMOND SIGMA T - ASTERISK
SIGMA Z - CROSS TEMPERATURE - SQUARE

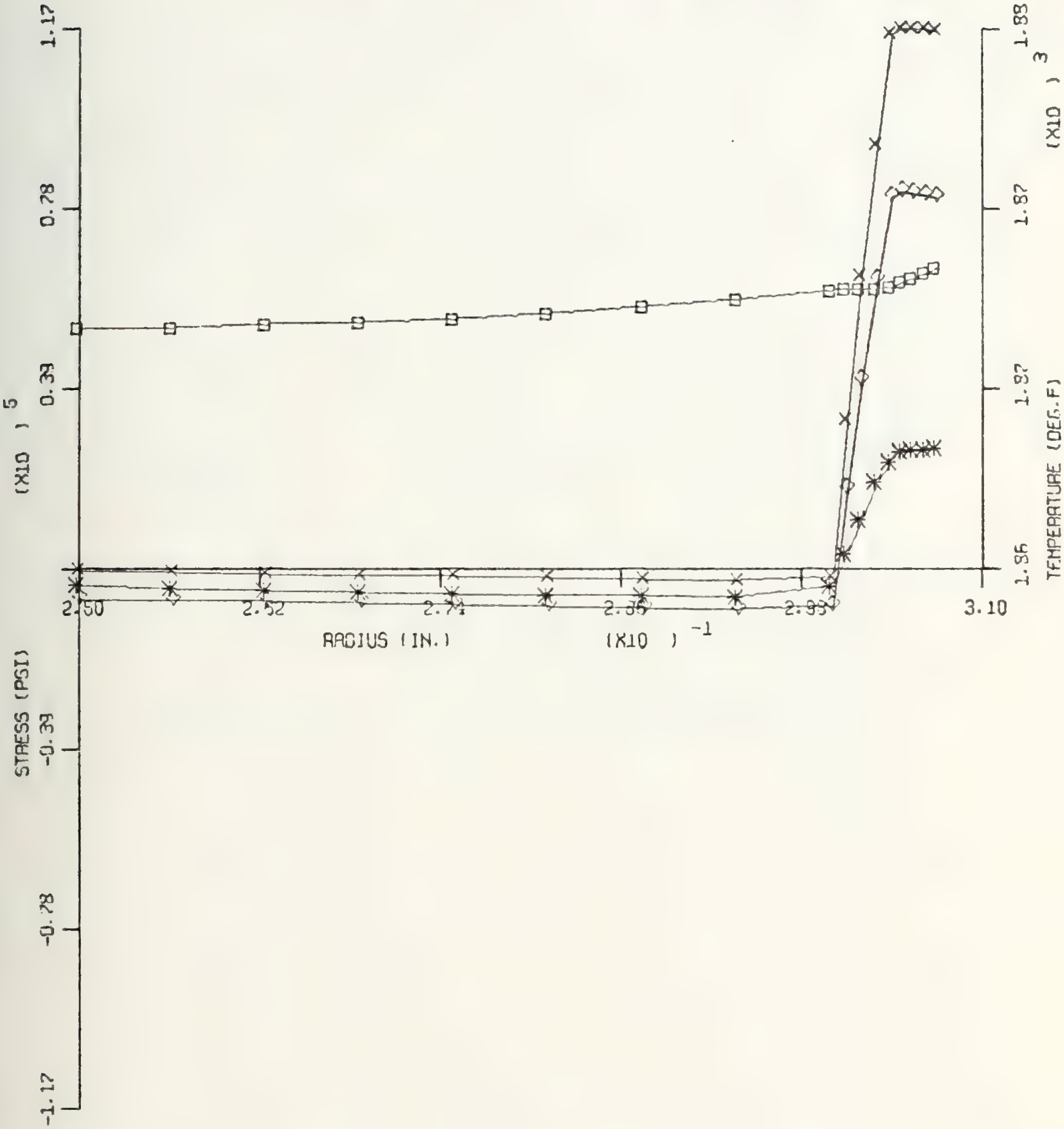
STRESS SCALE = 1.799X 10⁴ PSI/INCH

TEMPERATURE SCALE = 6.7 DEGREES/INCH

RADIAL POSITION SCALE = 0.01 INCHES/INCH

FIGURE 17(d)

STRESS AND TEMPERATURE
VERSUS RADIAL POSITION
TIME = 0.750 CONF.3-120



SIGMA R - DIAMOND SIGMA T - ASTERISK
SIGMA Z - CROSS TEMPERATURE - SQUARE

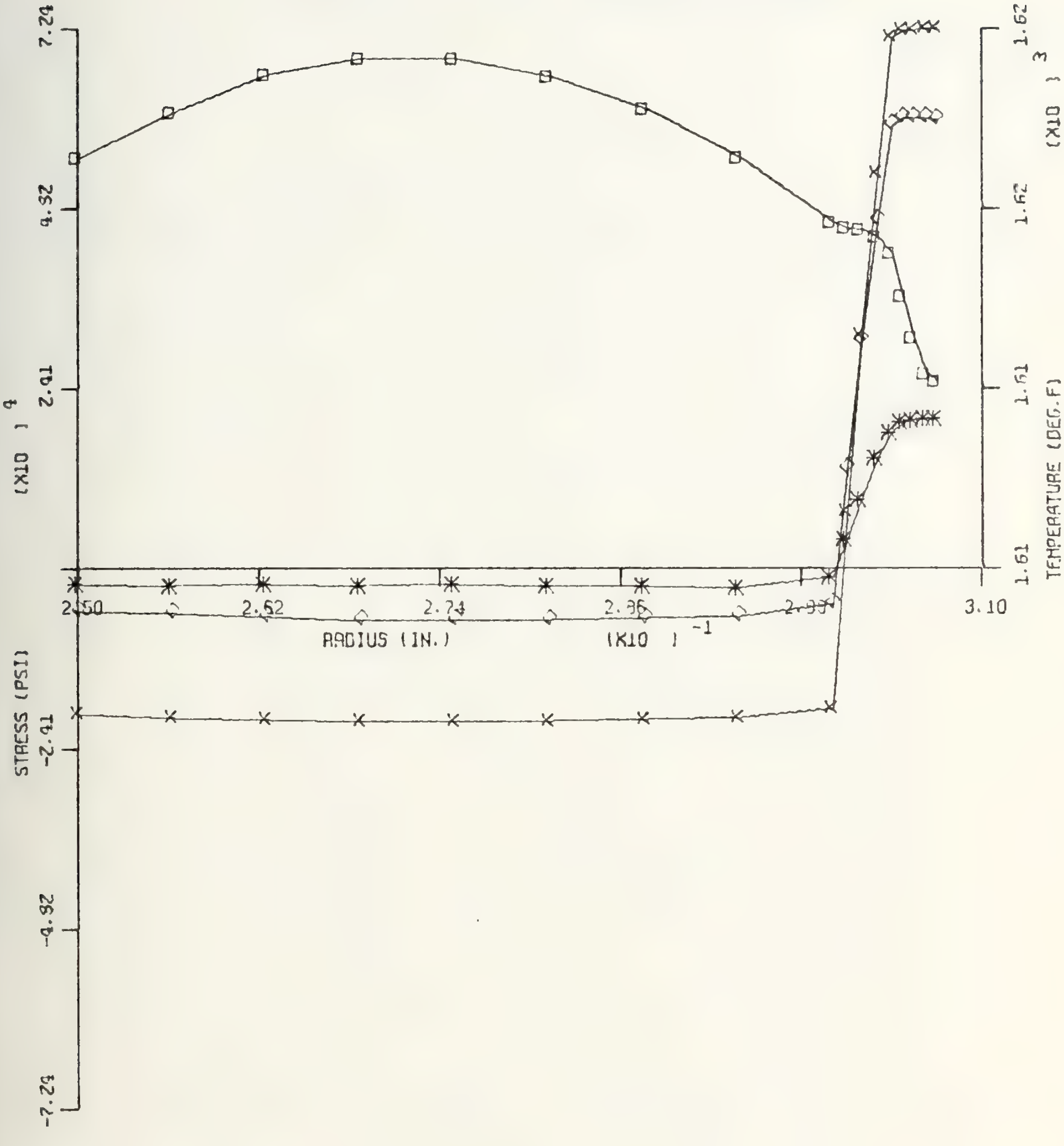
STRESS SCALE = 3.909X 10^4 PSI/INCH

TEMPERATURE SCALE = 6.7 DEGREES/INCH

RADIAL POSITION SCALE = 0.01 INCHES/INCH

FIGURE 17(e)

STRESS AND TEMPERATURE
VERSUS RADIAL POSITION
TIME = 0.800 CONF.3-120



SIGMA R - DIAMOND SIGMA T - ASTERISK
SIGMA Z - CROSS TEMPERATURE - SQUARE

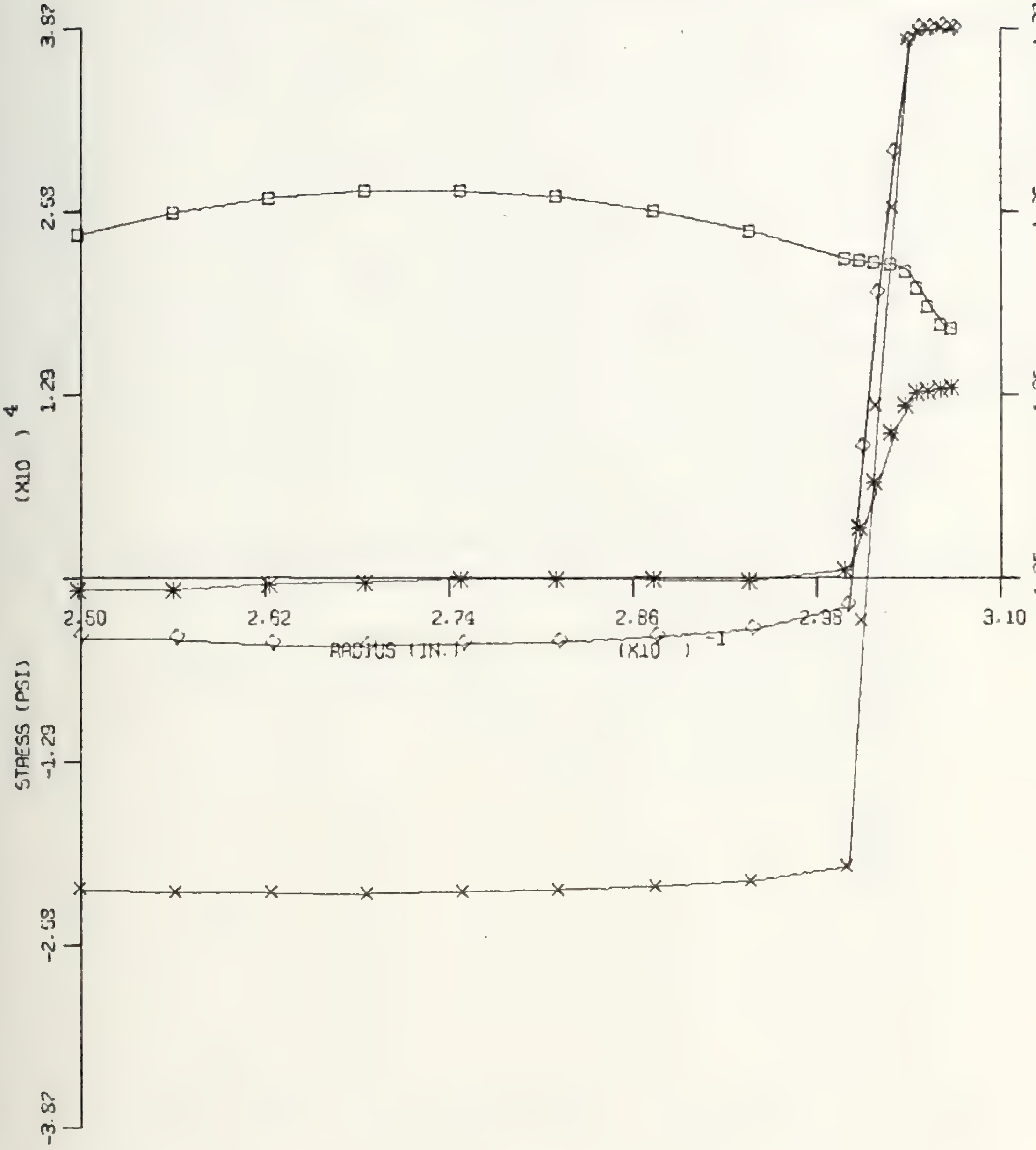
STRESS SCALE = 2.412X 10^4 PSI/INCH

TEMPERATURE SCALE = 3.3 DEGREES/INCH

RADIAL POSITION SCALE = 0.01 INCHES/INCH

FIGURE 17(f)

STRESS AND TEMPERATURE
VERSUS RADIAL POSITION
TIME = 0.850 CONF.3-120



SIGMA R - DIAMOND SIGMA T - ASTERISK
SIGMA Z - CROSS TEMPERATURE - SQUARE

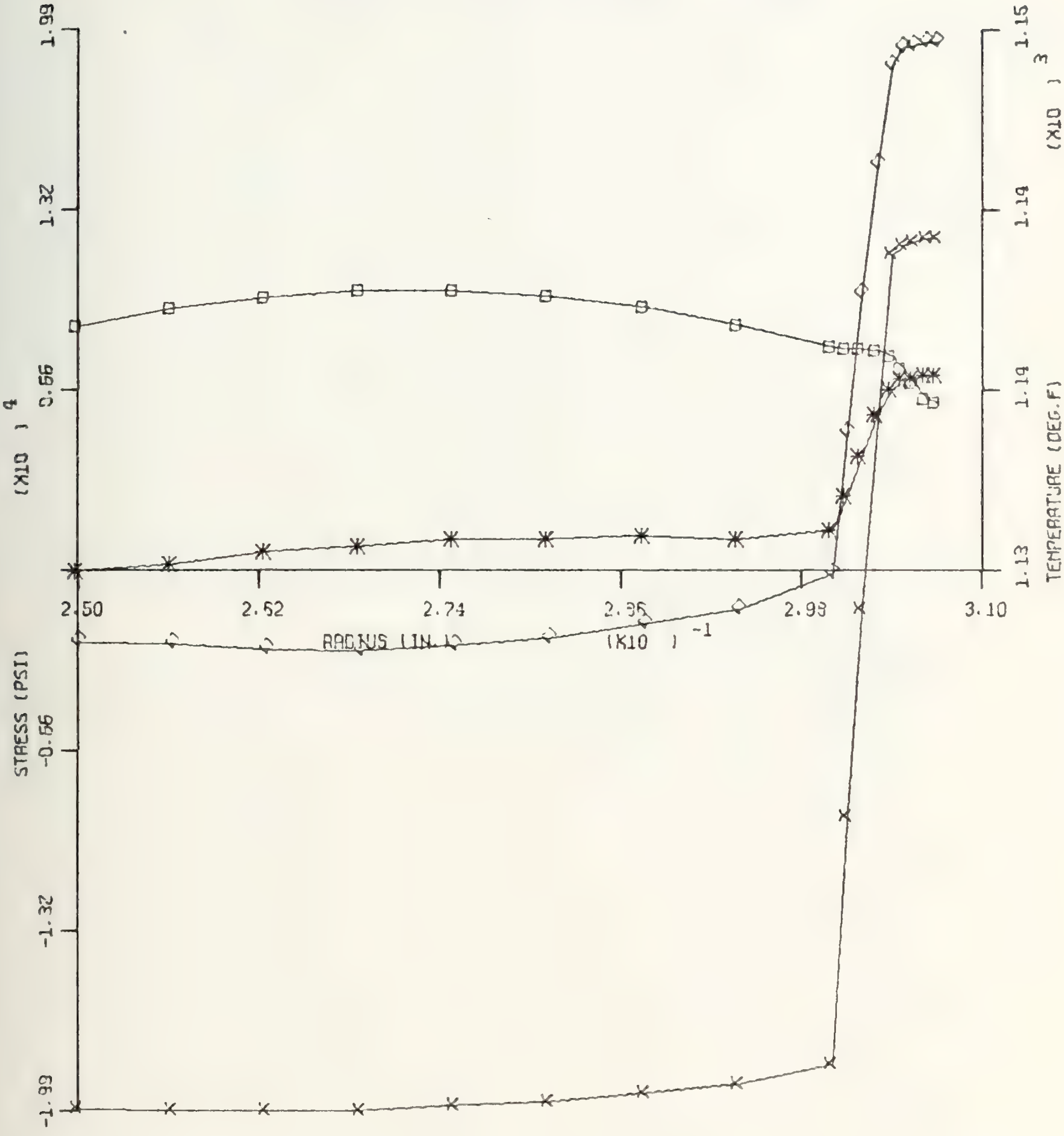
STRESS SCALE = 1.290X 10^4 PSI/INCH

TEMPERATURE SCALE = 6.7 DEGREES/INCH

RADIAL POSITION SCALE = 0.01 INCHES/INCH

FIGURE 17(g)

STRESS AND TEMPERATURE
VERSUS RADIAL POSITION
TIME = 0.900 CONF.3-120



SIGMA R - DIAMOND SIGMA T - ASTERISK
SIGMA Z - CROSS TEMPERATURE - SQUARE

STRESS SCALE = 5.522X 10^3 PSI/INCH

TEMPERATURE SCALE = 6.7 DEGREES/INCH

RADIAL POSITION SCALE = 0.01 INCHES/INCH

FIGURE 17(h)

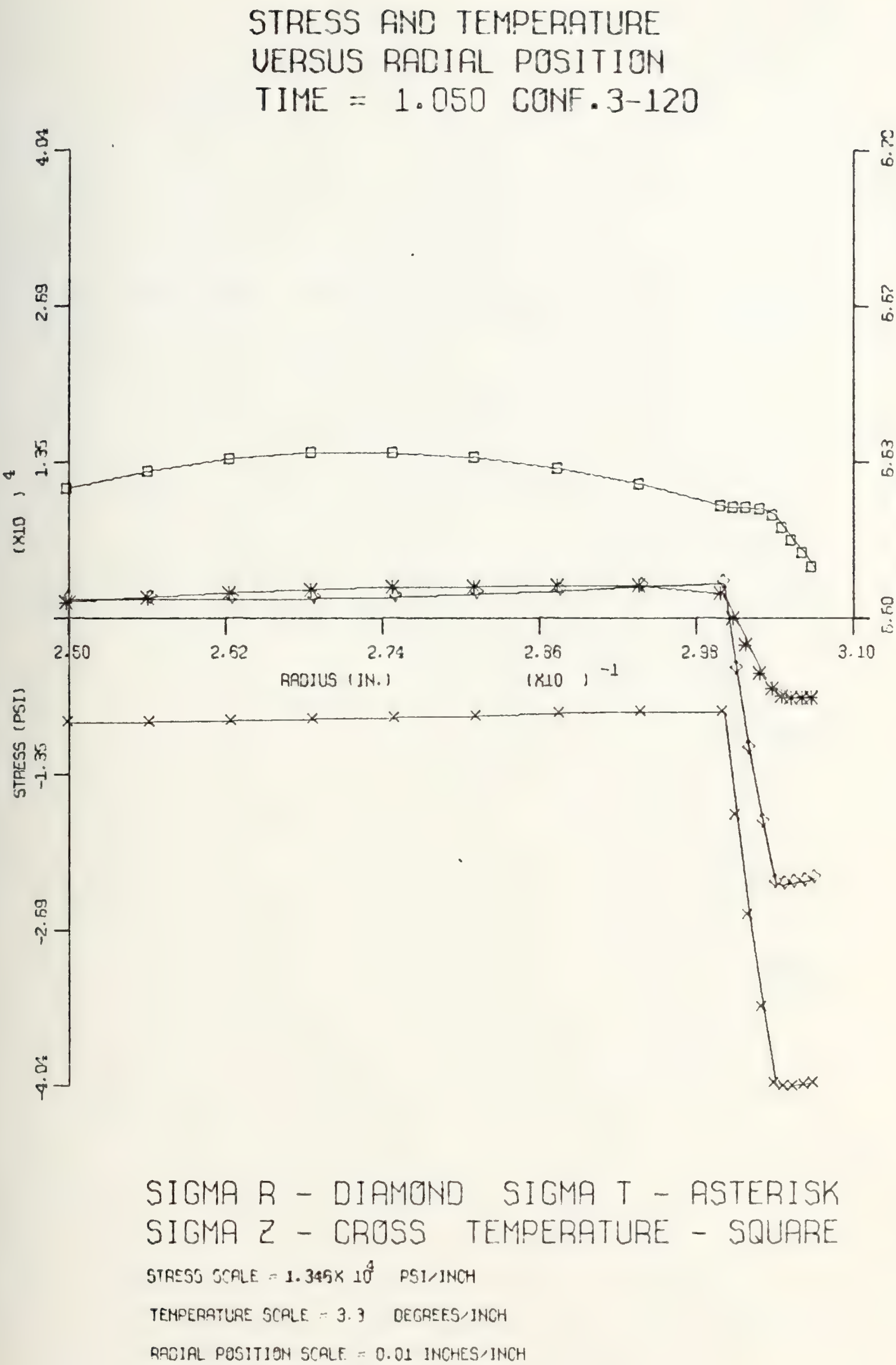
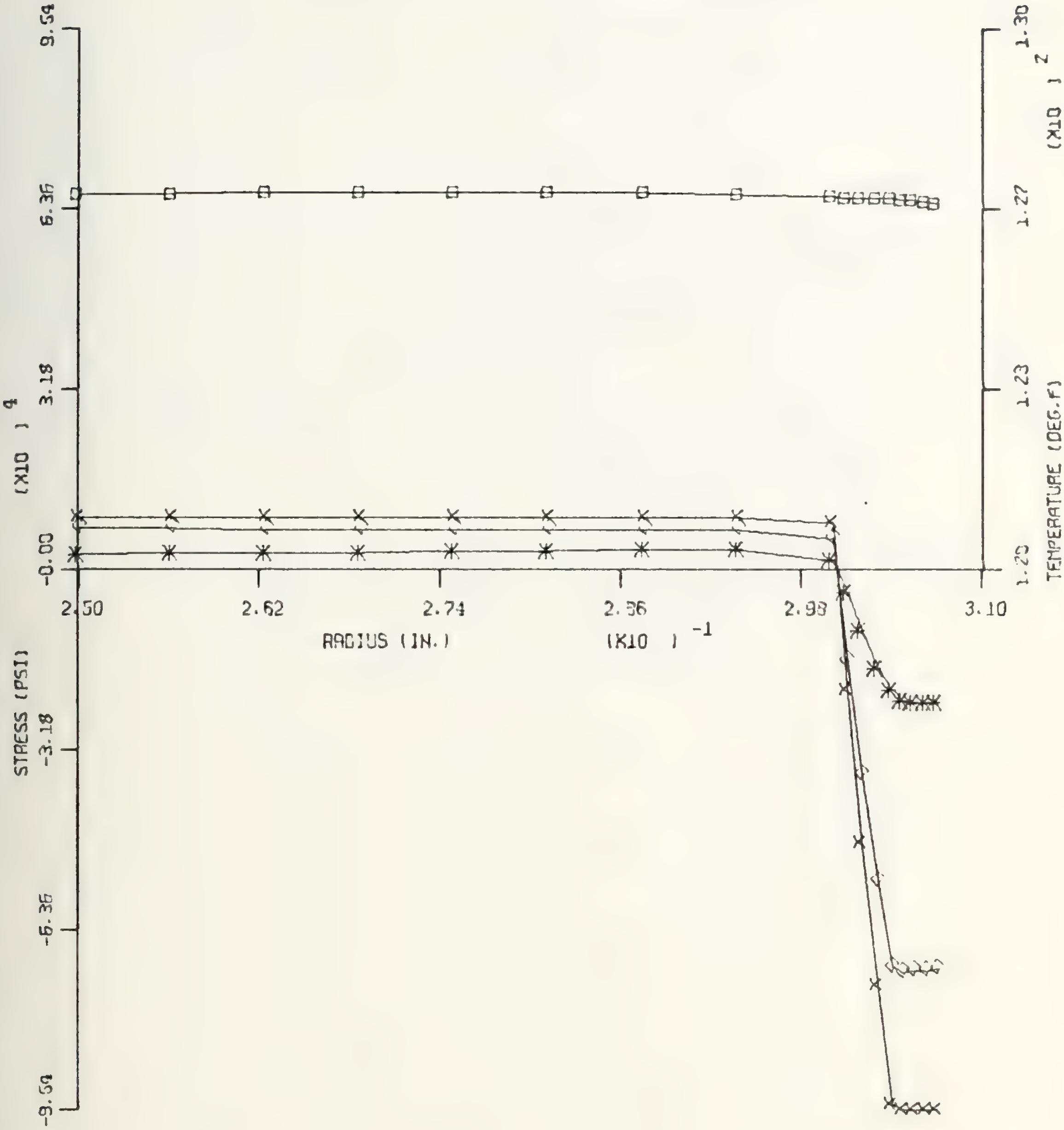

STRESS AND TEMPERATURE
VERSUS RADIAL POSITION
TIME = 1.050 CONF.3-120
4.04
2.69
1.35
-1.35
-2.69
-4.04
(X10) 4
STRESS (PSI)
6.70
6.67
6.63
6.60
(X10) 2
TEMPERATURE (DEG.F)
2.50
2.62
2.74
2.86
2.98
3.10
RADIUS (IN.)
(X10) -1
SIGMA R - DIAMOND SIGMA T - ASTERISK
SIGMA Z - CROSS TEMPERATURE - SQUARE
STRESS SCALE = 1.346X 10^4 PSI/INCH
TEMPERATURE SCALE = 3.3 DEGREES/INCH
RADIAL POSITION SCALE = 0.01 INCHES/INCH

FIGURE 17(i)

STRESS AND TEMPERATURE
VERSUS RADIAL POSITION
TIME = 1.650 CONF.3-120



SIGMA R - DIAMOND SIGMA T - ASTERISK
SIGMA Z - CROSS TEMPERATURE - SQUARE

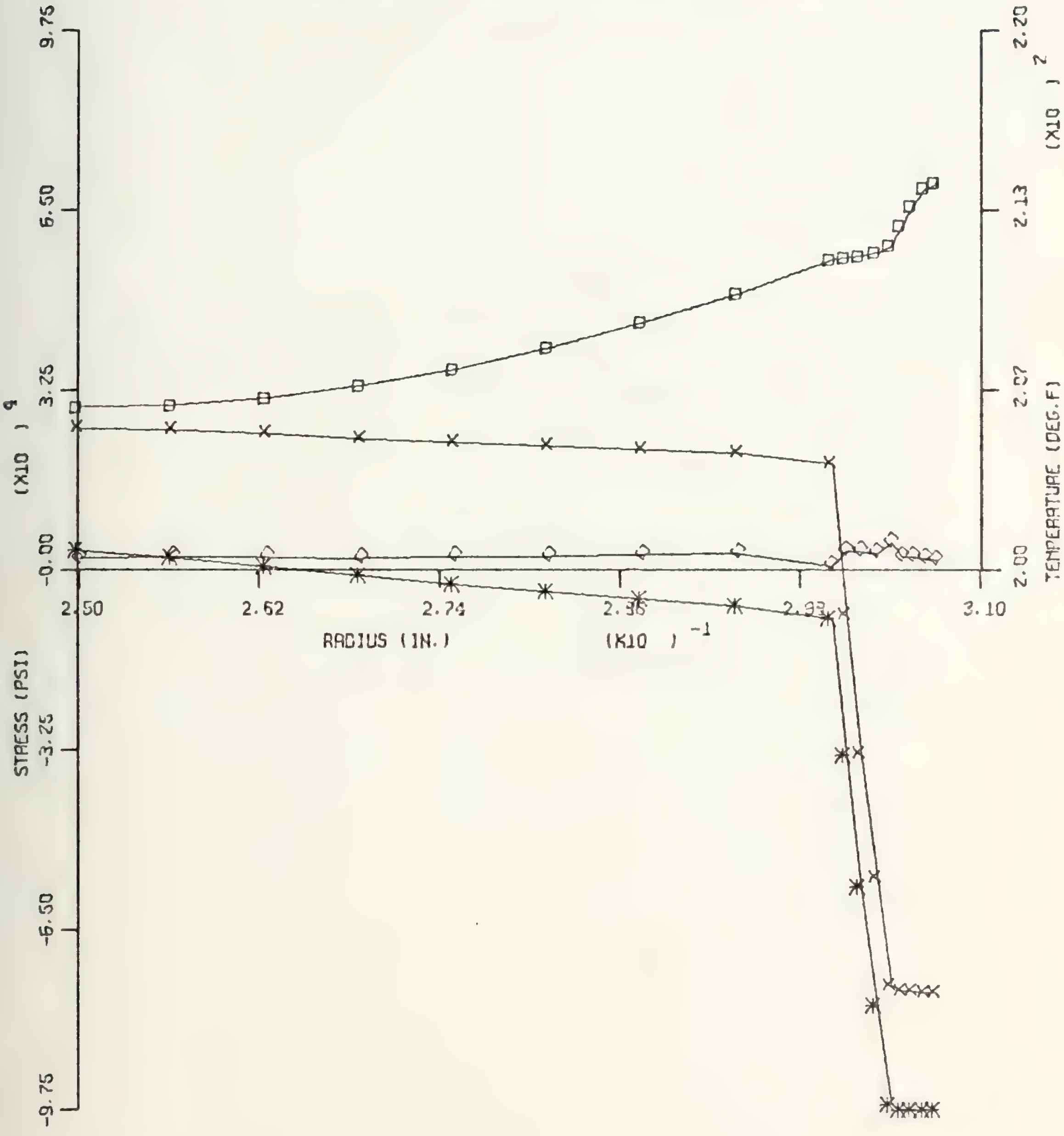
STRESS SCALE = 3.178X 10^4 PSI/INCH

TEMPERATURE SCALE = 3.3 DEGREES/INCH

RADIAL POSITION SCALE = 0.01 INCHES/INCH

FIGURE 17(j)

STRESS AND TEMPERATURE
VERSUS RADIAL POSITION
TIME = 0.050 CONF.3-180



SIGMA R - DIAMOND SIGMA T - ASTERISK
SIGMA Z - CROSS TEMPERATURE - SQUARE

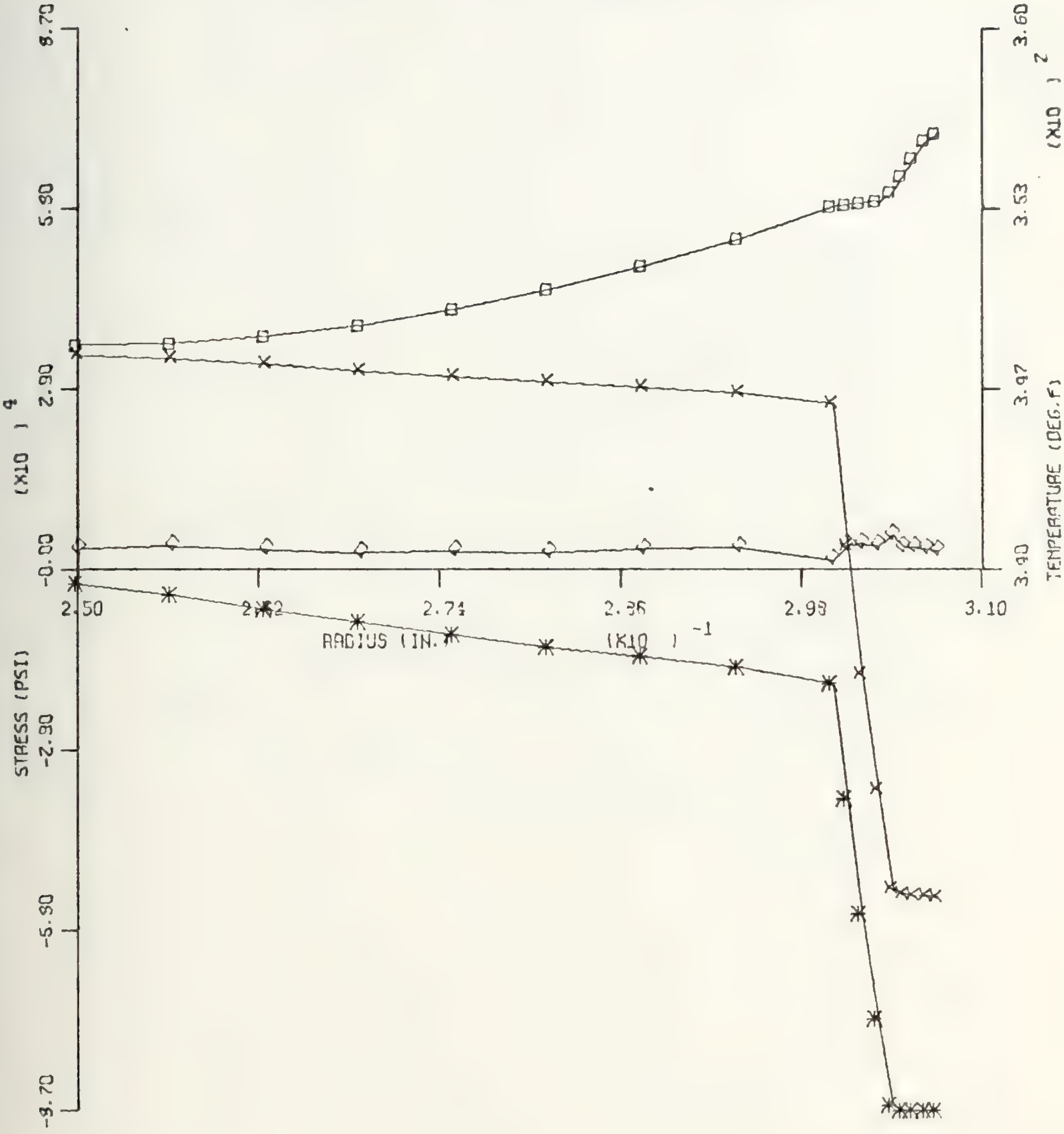
STRESS SCALE = 3.249K 10^4 PSI/INCH

TEMPERATURE SCALE = 6.7 DEGREES/INCH

RADIAL POSITION SCALE = 0.01 INCHES/INCH

FIGURE 18(a)

STRESS AND TEMPERATURE
VERSUS RADIAL POSITION
TIME = 0.100 CONF.3-180



SIGMA R - DIAMOND SIGMA T - ASTERISK
SIGMA Z - CROSS TEMPERATURE - SQUARE

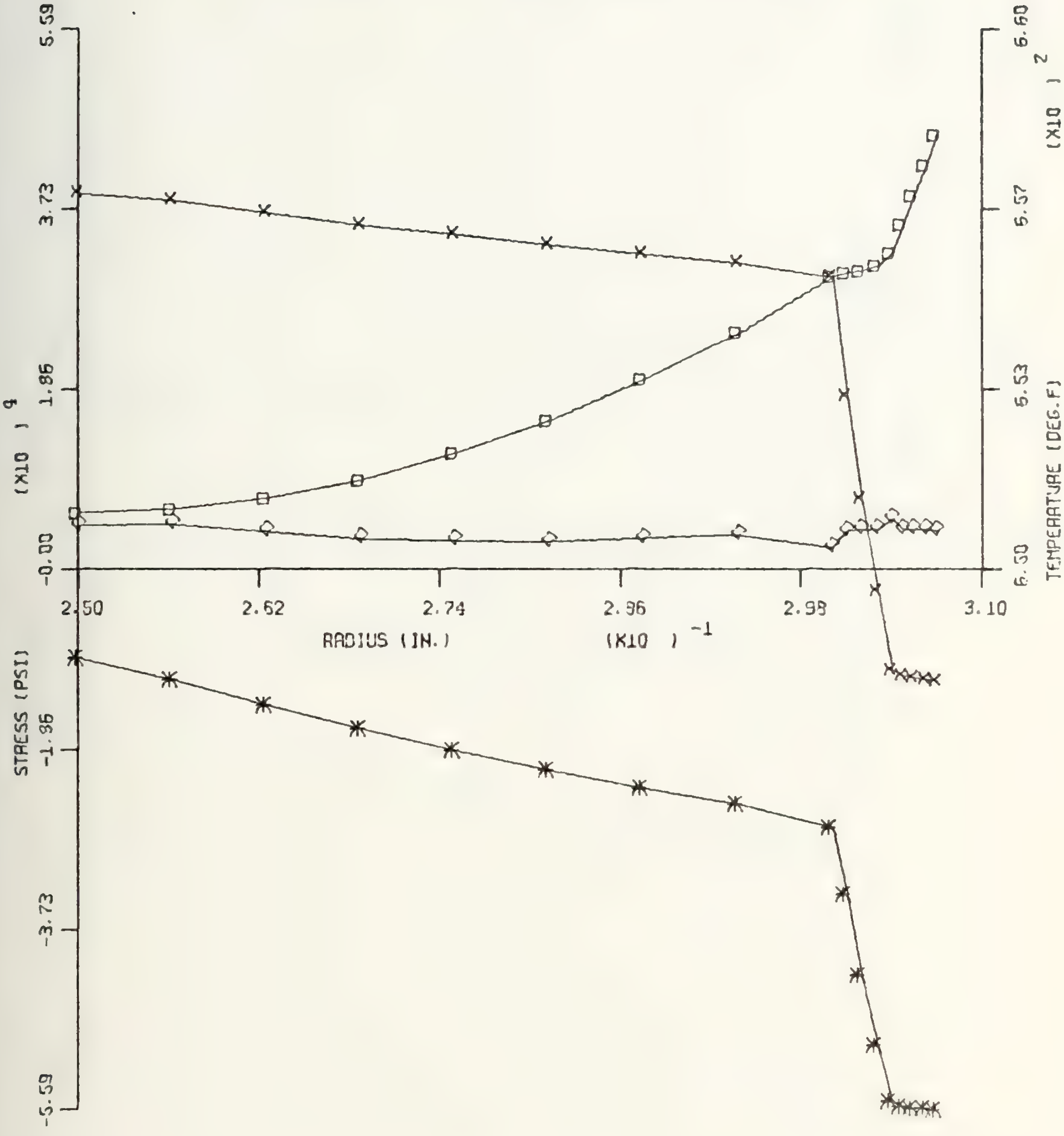
STRESS SCALE = 2.893X 10^4 PSI/INCH

TEMPERATURE SCALE = 5.7 DEGREES/INCH

RADIAL POSITION SCALE = 0.01 INCHES/INCH

FIGURE 18(b)

STRESS AND TEMPERATURE
VERSUS RADIAL POSITION
TIME = 0.200 CONF.3-180



SIGMA R - DIAMOND SIGMA T - ASTERISK
SIGMA Z - CROSS TEMPERATURE - SQUARE

STRESS SCALE = 1.865X 10^4 PSI/INCH

TEMPERATURE SCALE = 3.3 DEGREES/INCH

RADIAL POSITION SCALE = 0.01 INCHES/INCH

FIGURE 18(c)

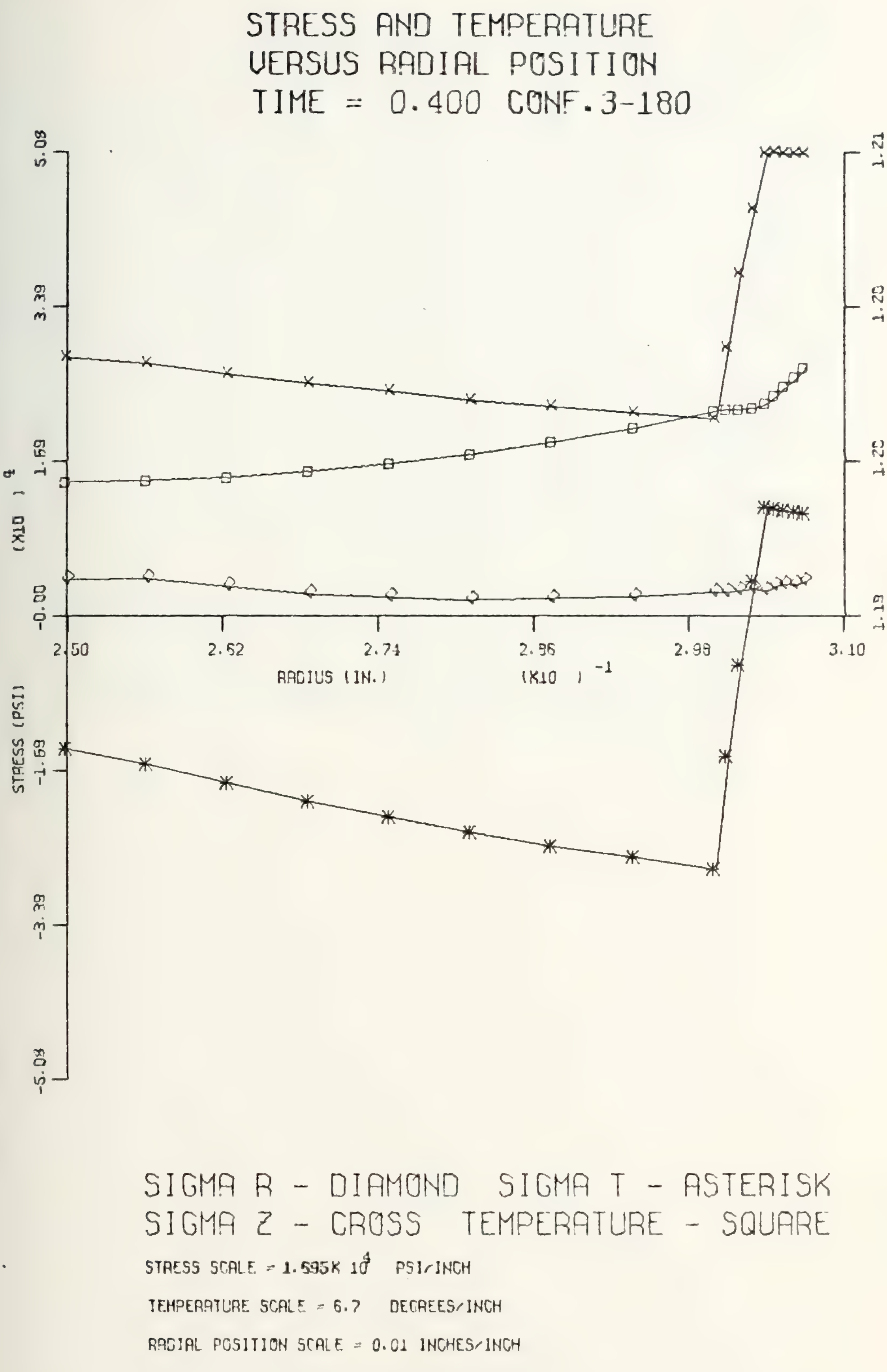
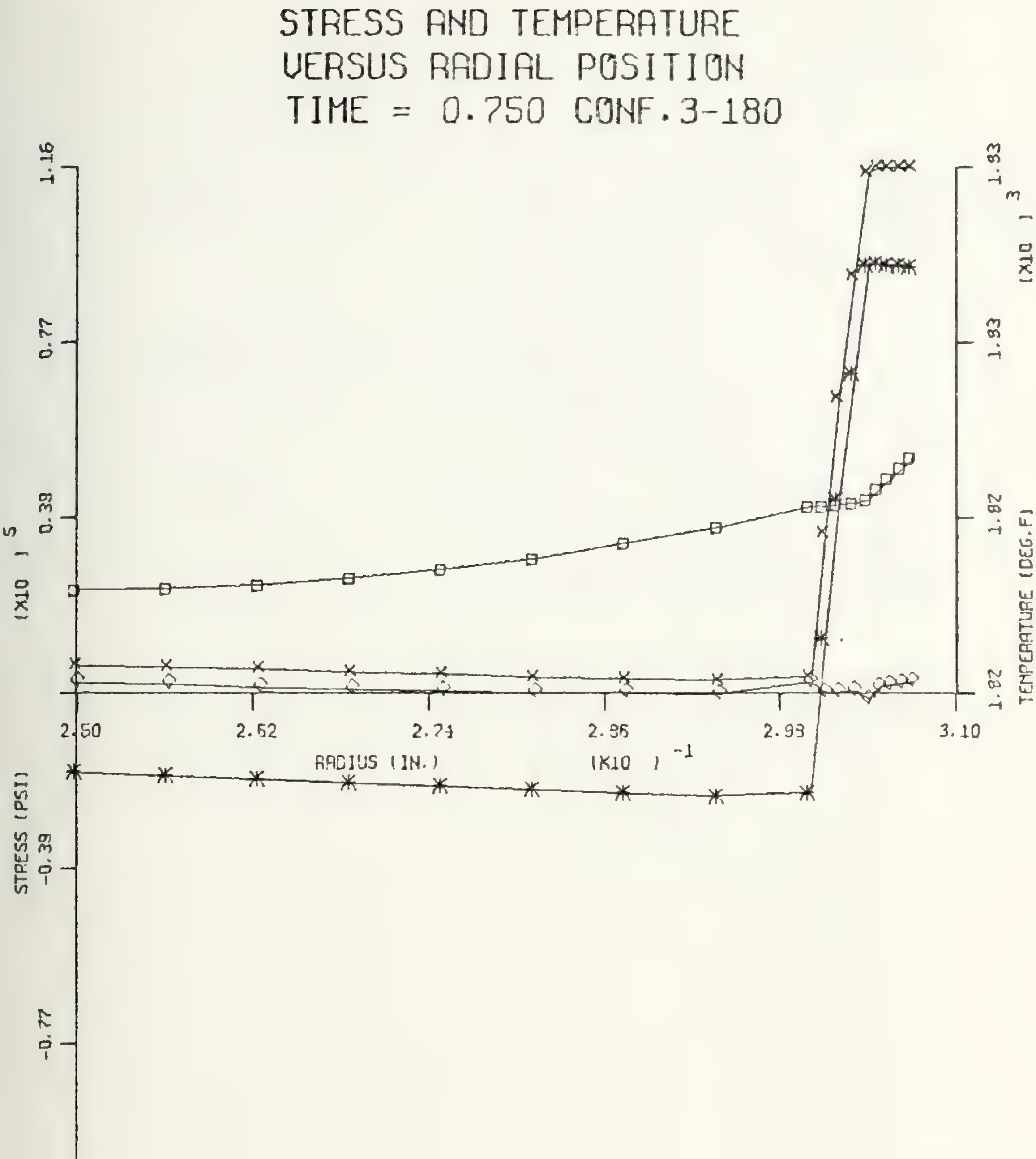

STRESS AND TEMPERATURE
VERSUS RADIAL POSITION
TIME = 0.400 CONF.3-180
5.08
3.39
1.69
-0.00
-1.69
-3.39
-5.08
STRESS (PSI)
(X10) 4
1.21
1.20
1.20
1.19
(X10) 3
TEMPERATURE (DEG.F)
2.50
2.62
2.74
2.86
2.98
3.10
RADIUS (IN.)
(X10) -1
SIGMA R - DIAMOND SIGMA T - ASTERISK
SIGMA Z - CROSS TEMPERATURE - SQUARE
STRESS SCALE = 1.695X 10 4 PSI/INCH
TEMPERATURE SCALE = 6.7 DEGREES/INCH
RADIAL POSITION SCALE = 0.01 INCHES/INCH

FIGURE 18(d)



SIGMA R - DIAMOND SIGMA T - ASTERISK
SIGMA Z - CROSS TEMPERATURE - SQUARE

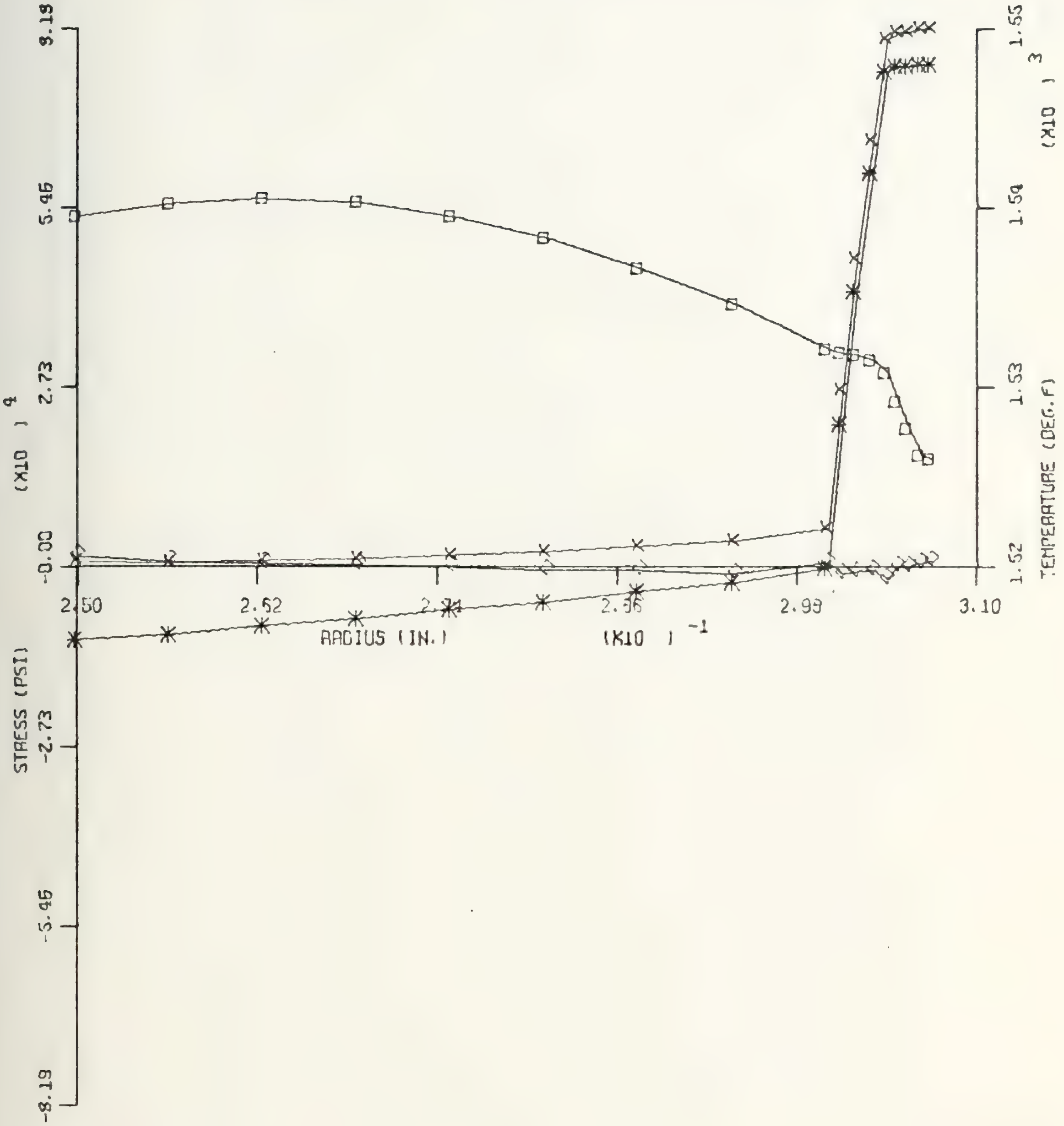
STRESS SCALE = 3.852X 10^4 PSI/INCH

TEMPERATURE SCALE = 3.3 DEGREES/INCH

RADIAL POSITION SCALE = 0.01 INCHES/INCH

FIGURE 18(e)

STRESS AND TEMPERATURE
VERSUS RADIAL POSITION
TIME = 0.800 CONF.3-180



SIGMA R - DIAMOND SIGMA T - ASTERISK
SIGMA Z - CROSS TEMPERATURE - SQUARE

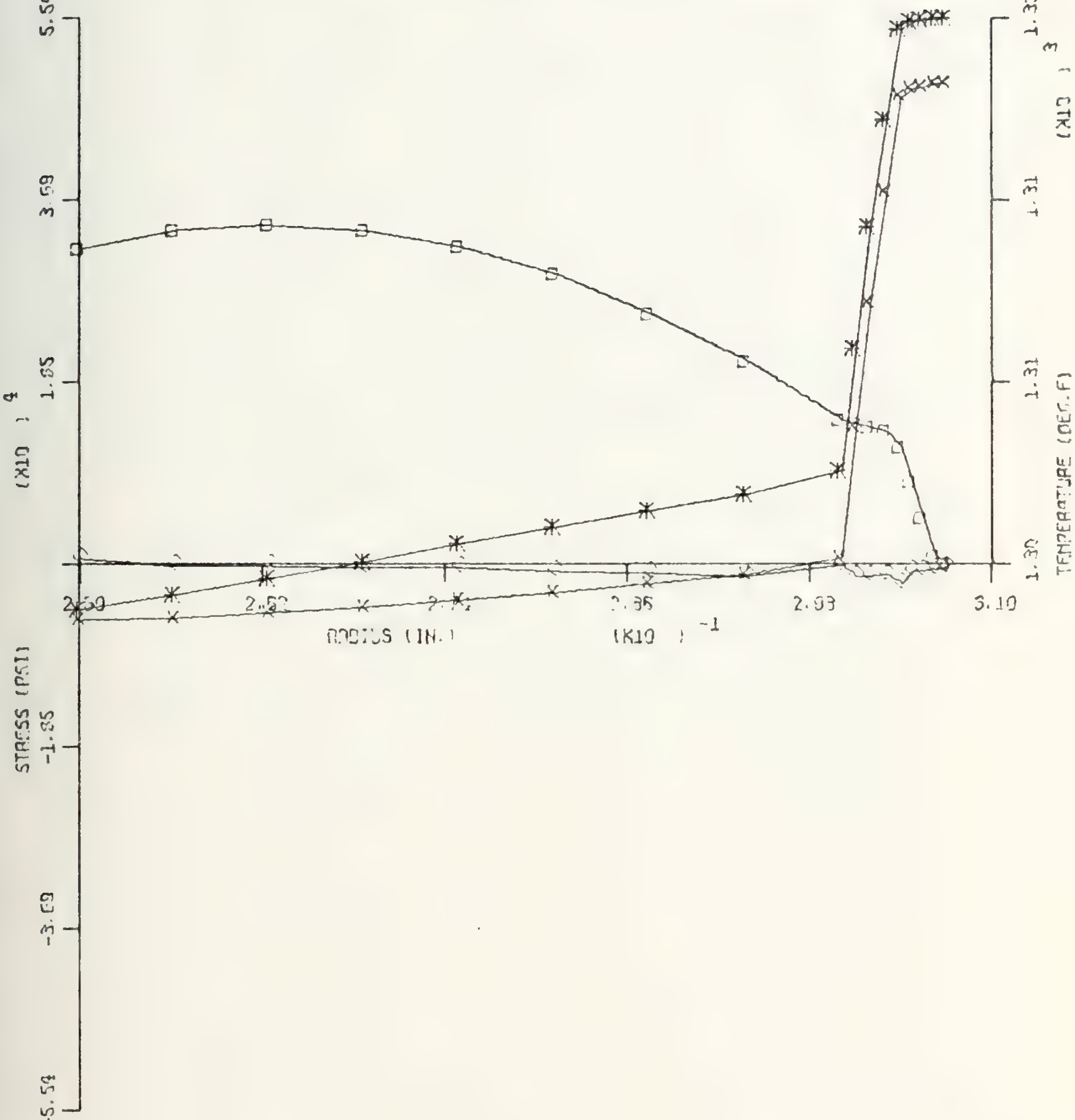
STRESS SCALE = 2.723×10^4 PSI/INCH

TEMPERATURE SCALE = 10.0 DEGREES/INCH

RADIAL POSITION SCALE = 0.01 INCHES/INCH

FIGURE 18(f)

STRESS AND TEMPERATURE
VERSUS RADIAL POSITION
TIME = 0.850 CONF.3-180



SIGMA R - DIAMOND SIGMA T - ASTERISK
SIGMA Z - CROSS TEMPERATURE - SQUARE

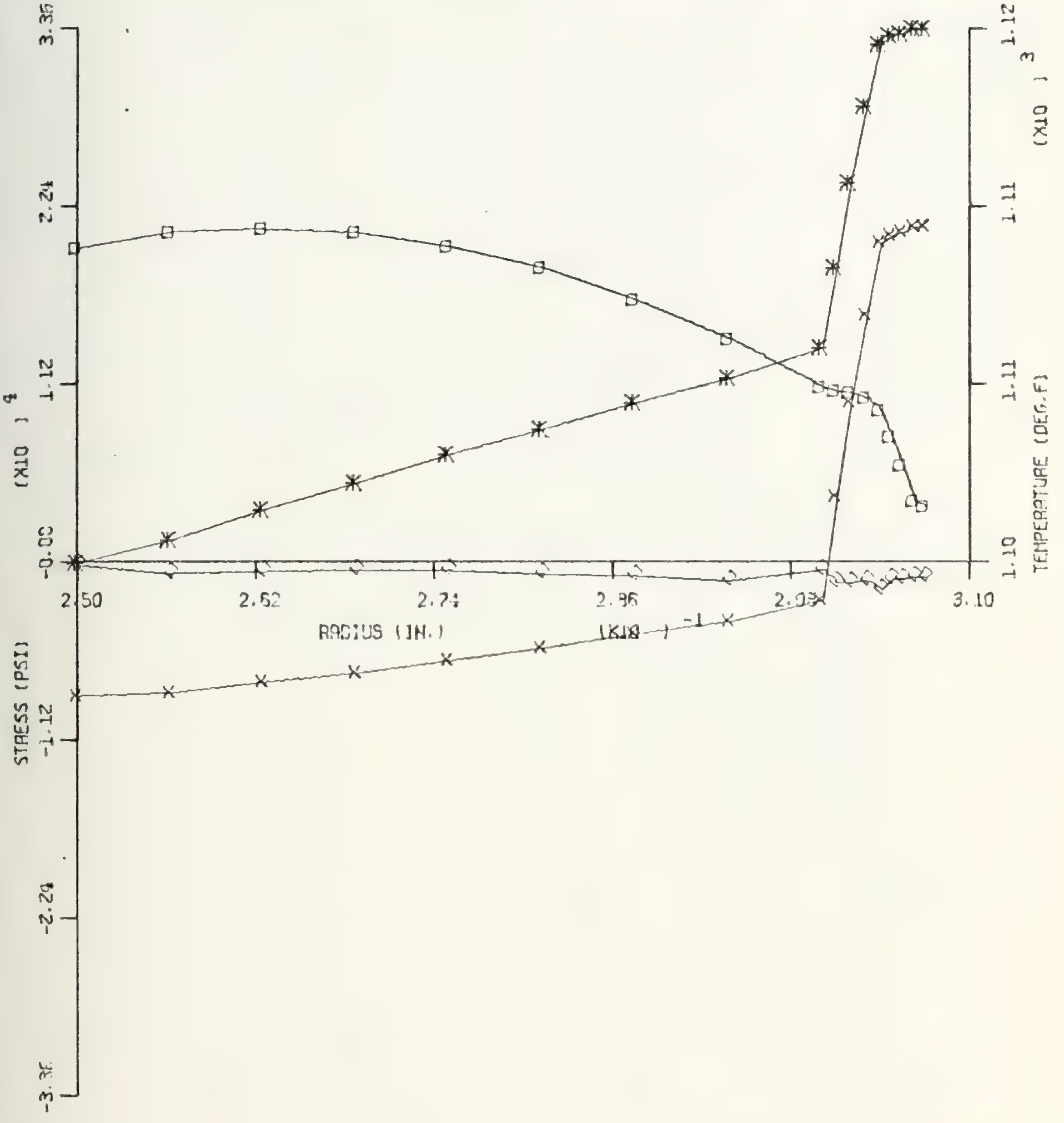
STRESS SCALE = 1.847X 10 PSI/INCH

TEMPERATURE SCALE = 6.7 DEGREES/INCH

RADIAL POSITION SCALE = 0.01 INCHES/INCH

FIGURE 18(g)

STRESS AND TEMPERATURE
VERSUS RADIAL POSITION
TIME = 0.900 CONF.3-180



SIGMA R - DIAMOND SIGMA T - ASTERISK
SIGMA Z - CROSS TEMPERATURE - SQUARE

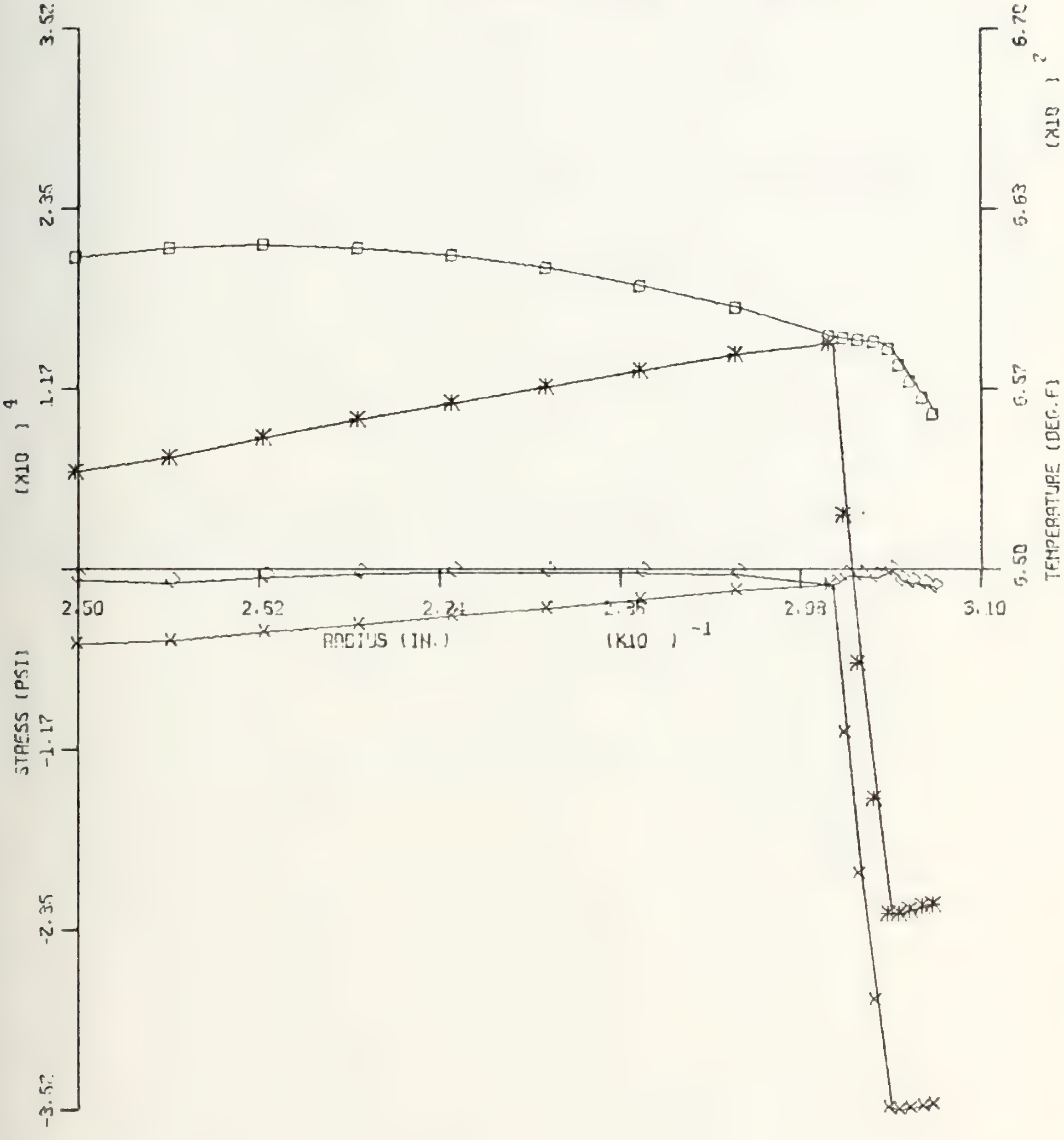
STRESS SCALE = 1.12 X 10^4 PSI/INCH

TEMPERATURE SCALE = 6.7 DEGREES/INCH

RADIAL POSITION SCALE = 0.01 INCHES/INCH

FIGURE 18(h)

STRESS AND TEMPERATURE
VERSUS RADIAL POSITION
TIME = 1.050 CONF.3-180



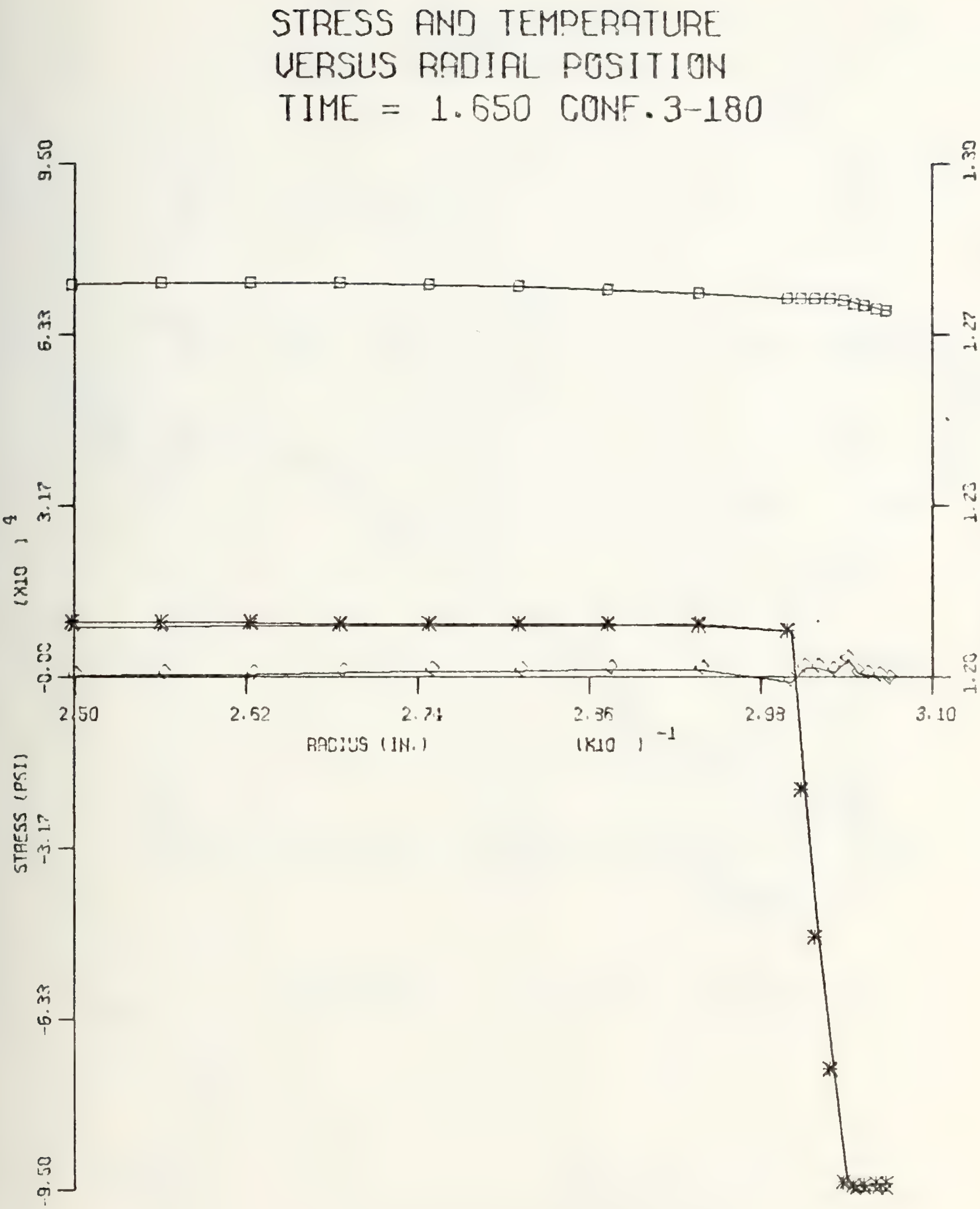
SIGMA R - DIAMOND SIGMA T - ASTERISK
SIGMA Z - CROSS TEMPERATURE - SQUARE

STRESS SCALE = 1.173X 10⁴ PSI/INCH

TEMPERATURE SCALE = 6.7 DEGREES/INCH

RADIAL POSITION SCALE = 0.01 INCHES/INCH

FIGURE 18(i)



SIGMA R - DIAMOND SIGMA T - ASTERISK
SIGMA Z - CROSS TEMPERATURE - SQUARE

STRESS SCALE = 3.157X 10^4 PSI/INCH

TEMPERATURE SCALE = 3.3 DEGREES/INCH

RADIAL POSITION SCALE = 0.01 INCHES/INCH

FIGURE 18(j)

STRESS AND TEMPERATURE
VERSUS
CIRCUMFERENTIAL POSITION

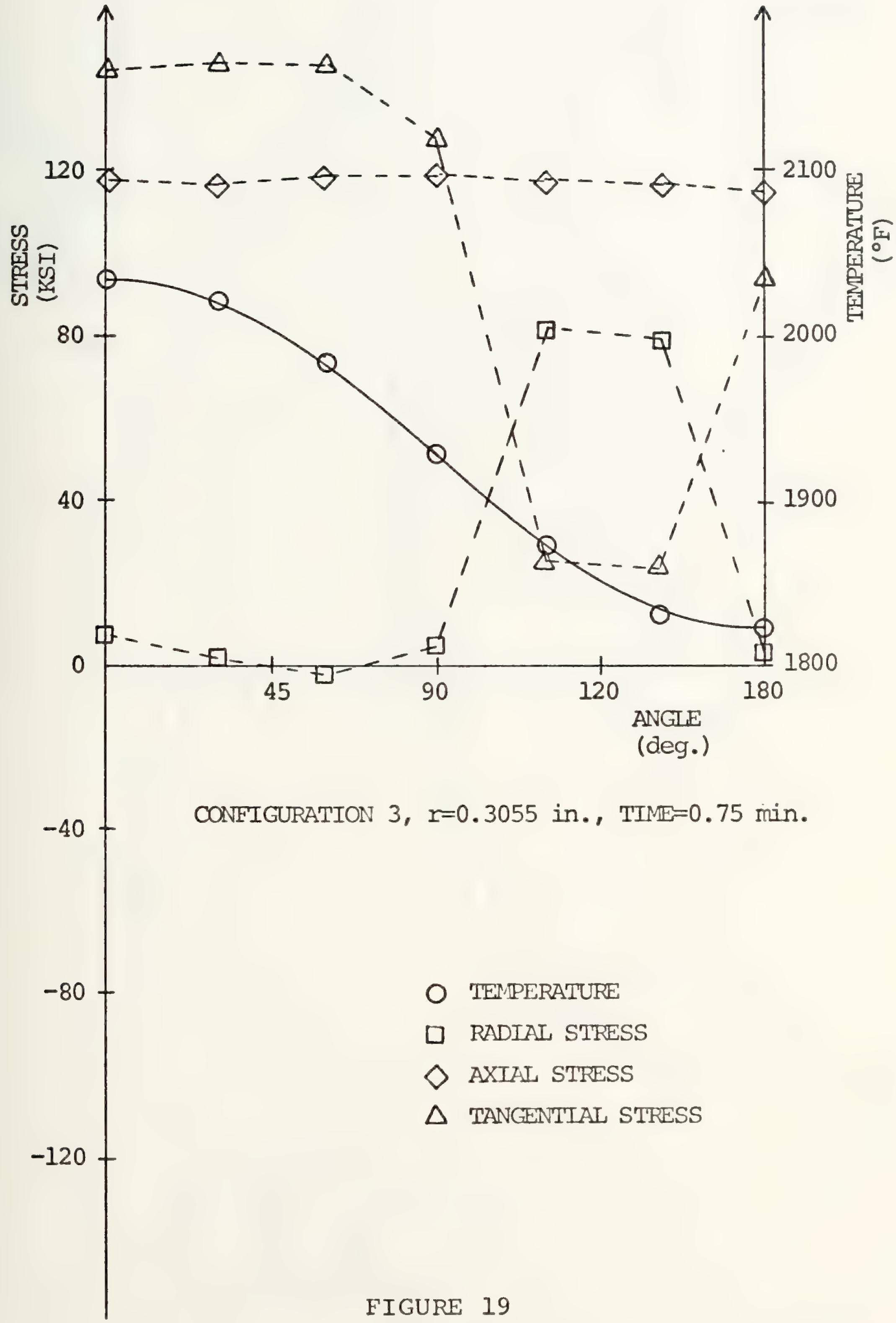


FIGURE 19

STRESS AND TEMPERATURE
VERSUS TIME

CONFIGURATION 3, r=0.3055 in., ANGLE=60°

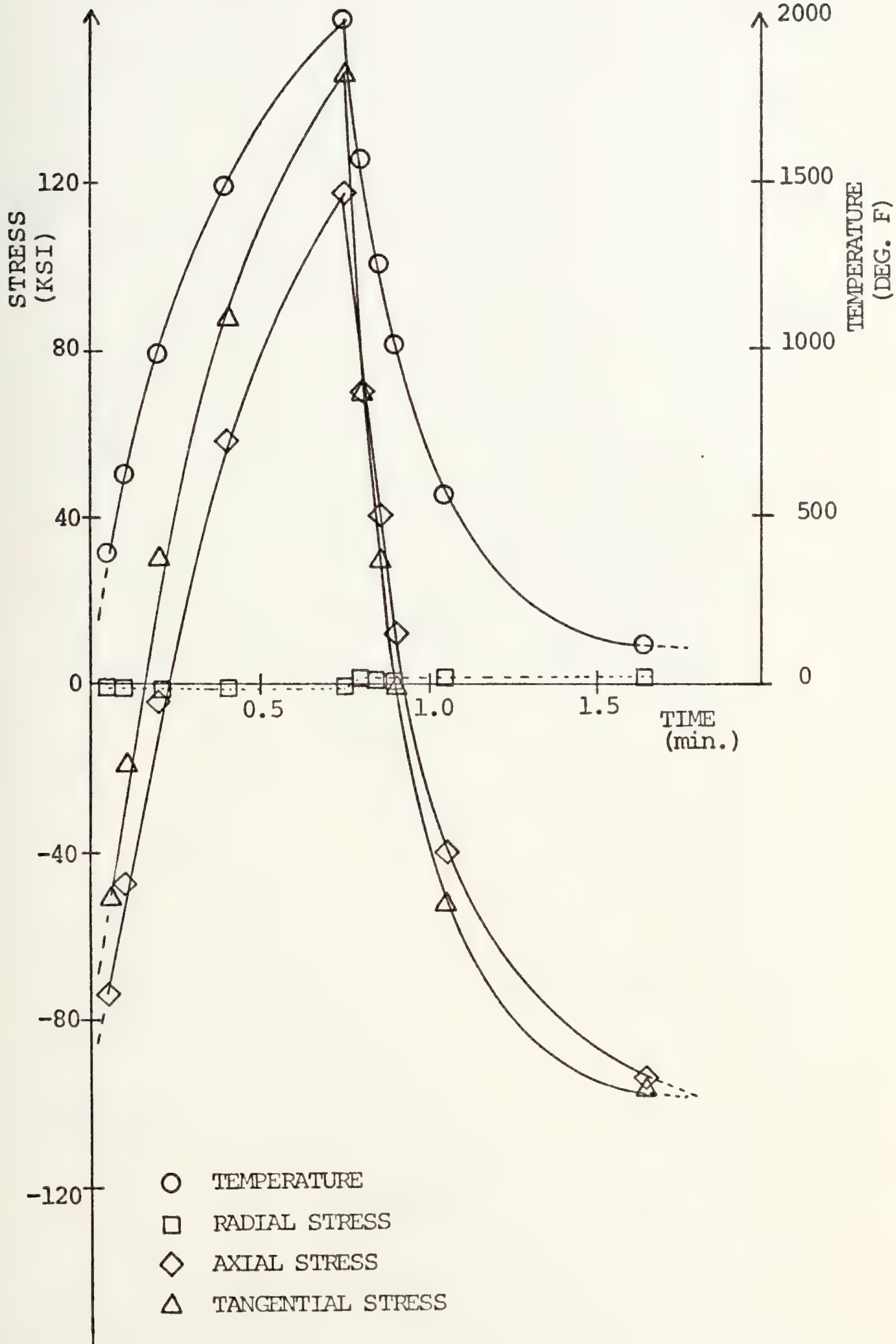


FIGURE 20

MAXIMUM AND MINIMUM PRINCIPAL STRESSES
VERSUS TIME
CONFIGURATION NUMBER THREE

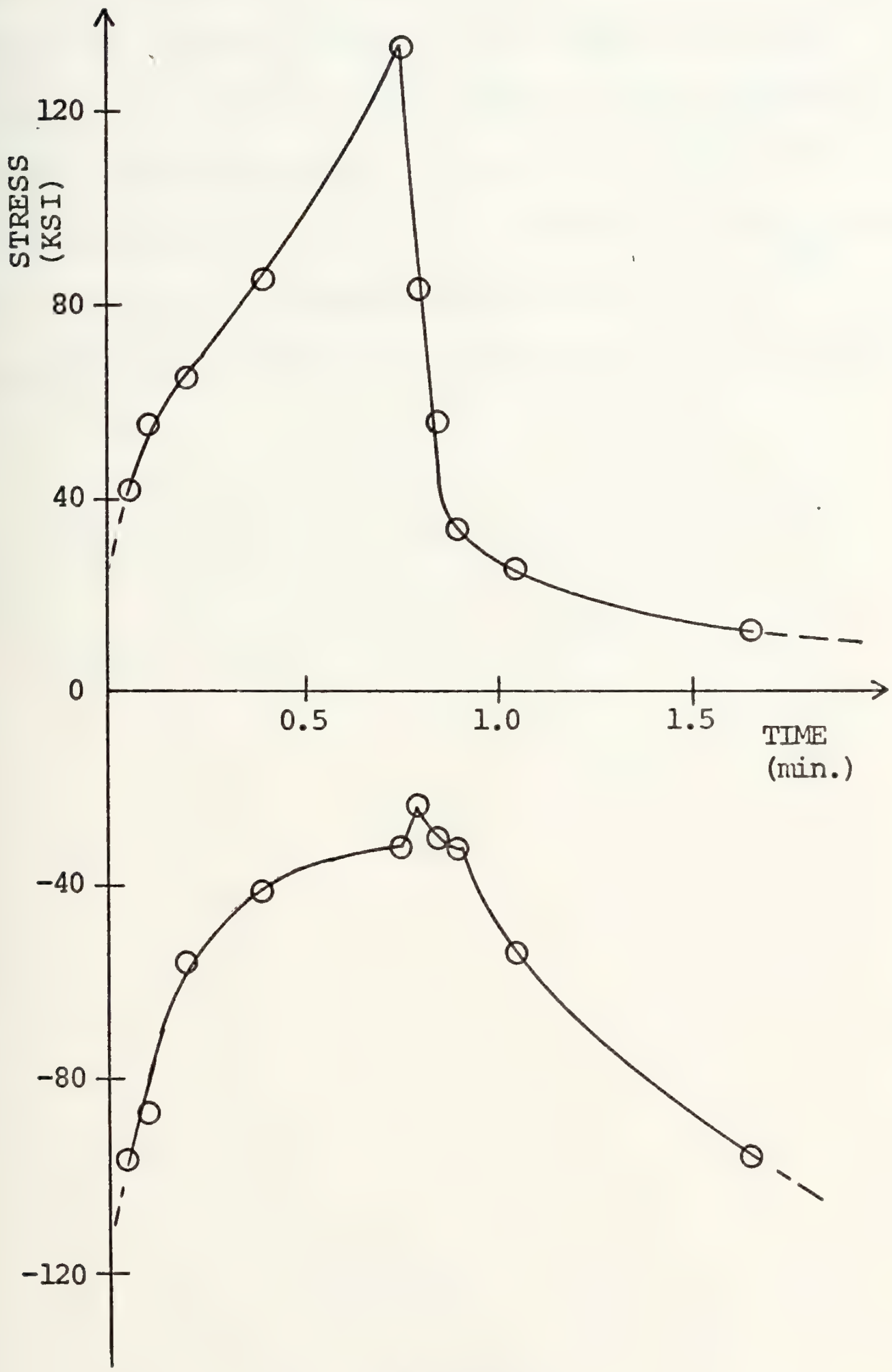
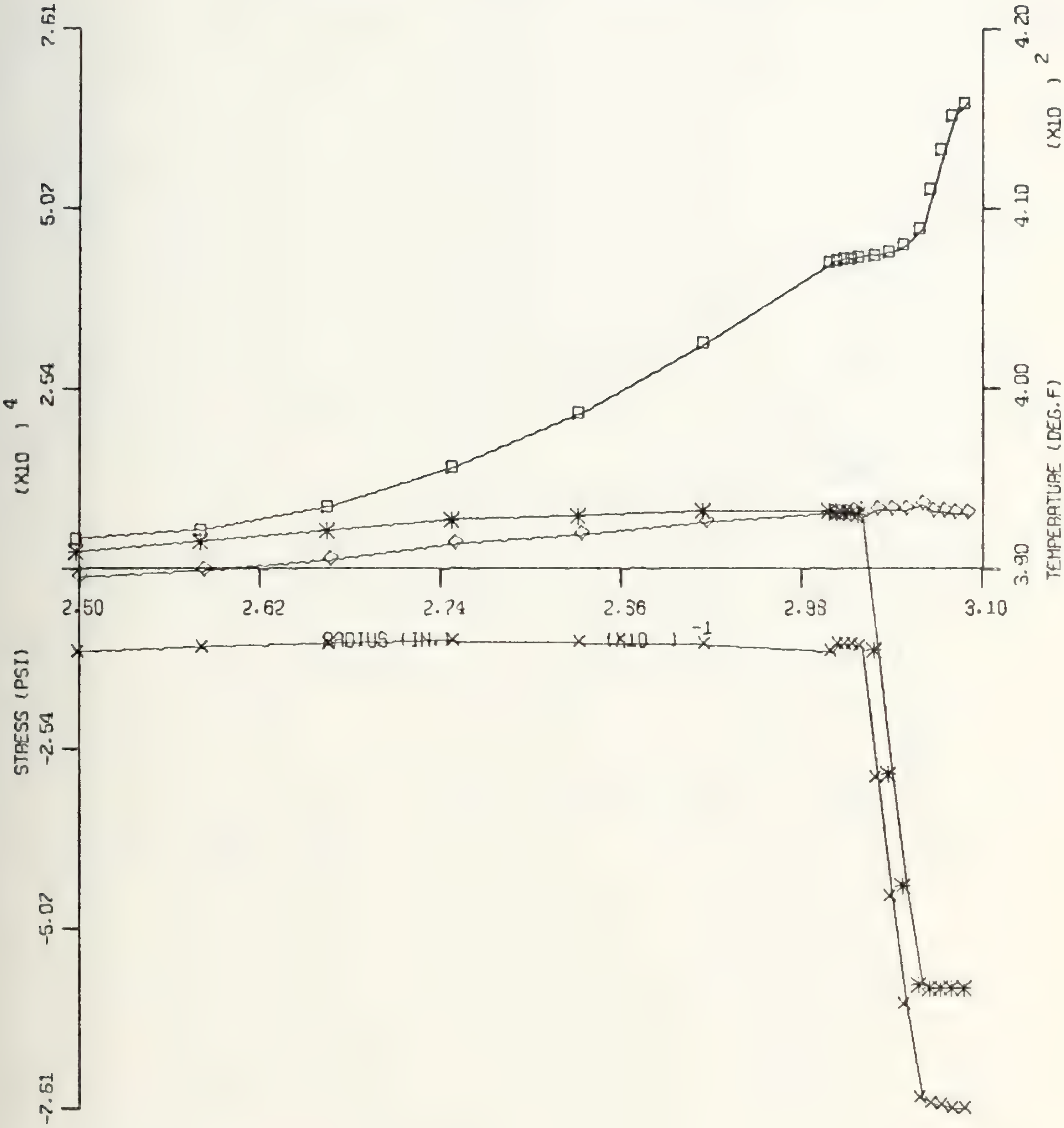


FIGURE 21

D. CONFIGURATION NUMBER FOUR

This coating arrangement included a layer of pure nickel between the substratum and the graded composition of configuration number three. The maximum principal stress (130.3 KSI) which occurred at the maximum temperature condition, was located at a radius of 0.3075 inches, the zero angle position, and at the midpoint of the cylinder length. As with the other configurations, the tangential and axial stresses were the largest for this configuration.

STRESS AND TEMPERATURE
VERSUS RADIAL POSITION
TIME = 0.050 CONF.4-0



SIGMA R - DIAMOND SIGMA T - ASTERISK
SIGMA Z - CROSS TEMPERATURE - SQUARE

STRESS SCALE = 2.537X 10^4 PSI/INCH

TEMPERATURE SCALE = 10.0 DEGREES/INCH

RADIAL POSITION SCALE = 0.01 INCHES/INCH

FIGURE 22(a)

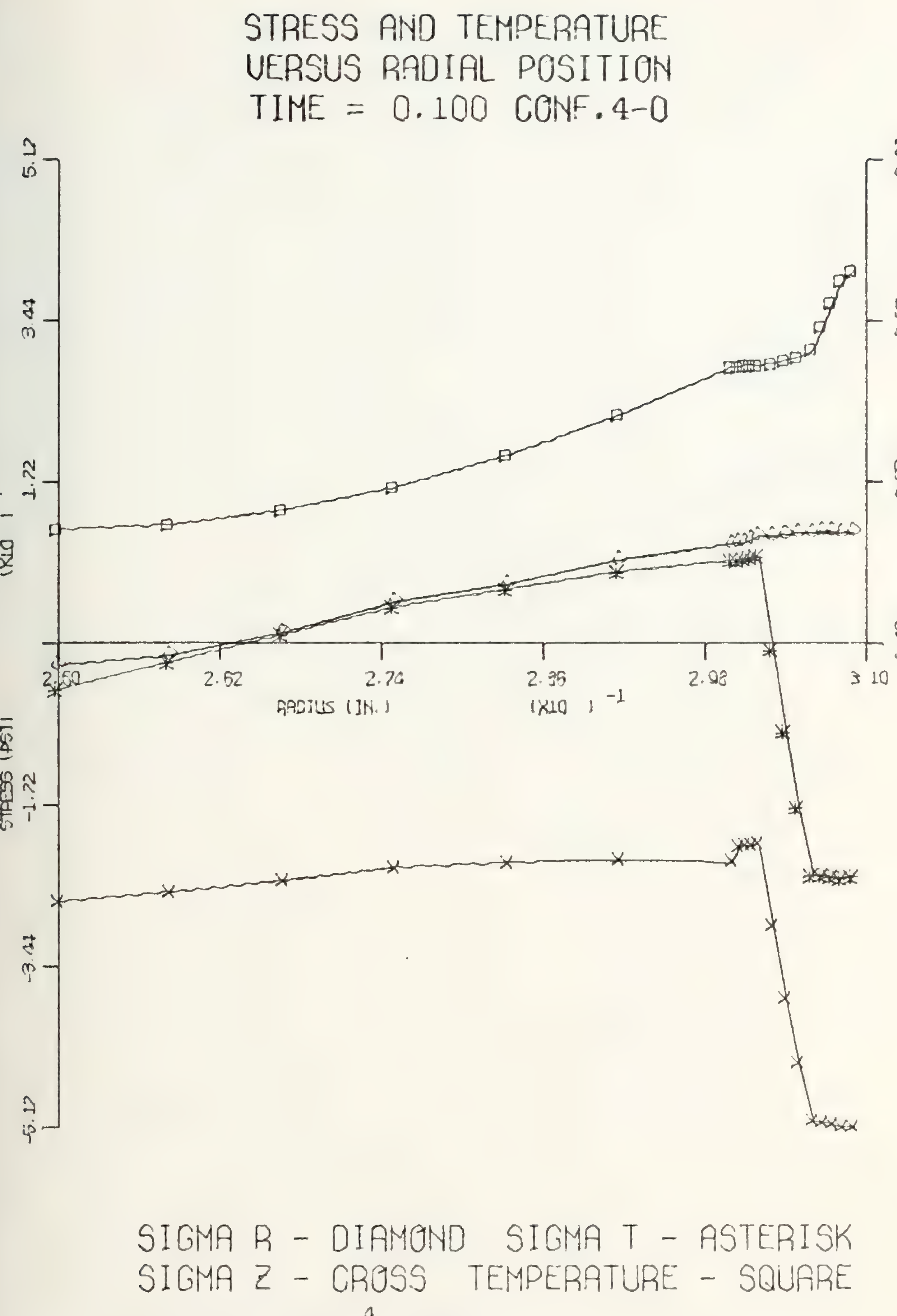
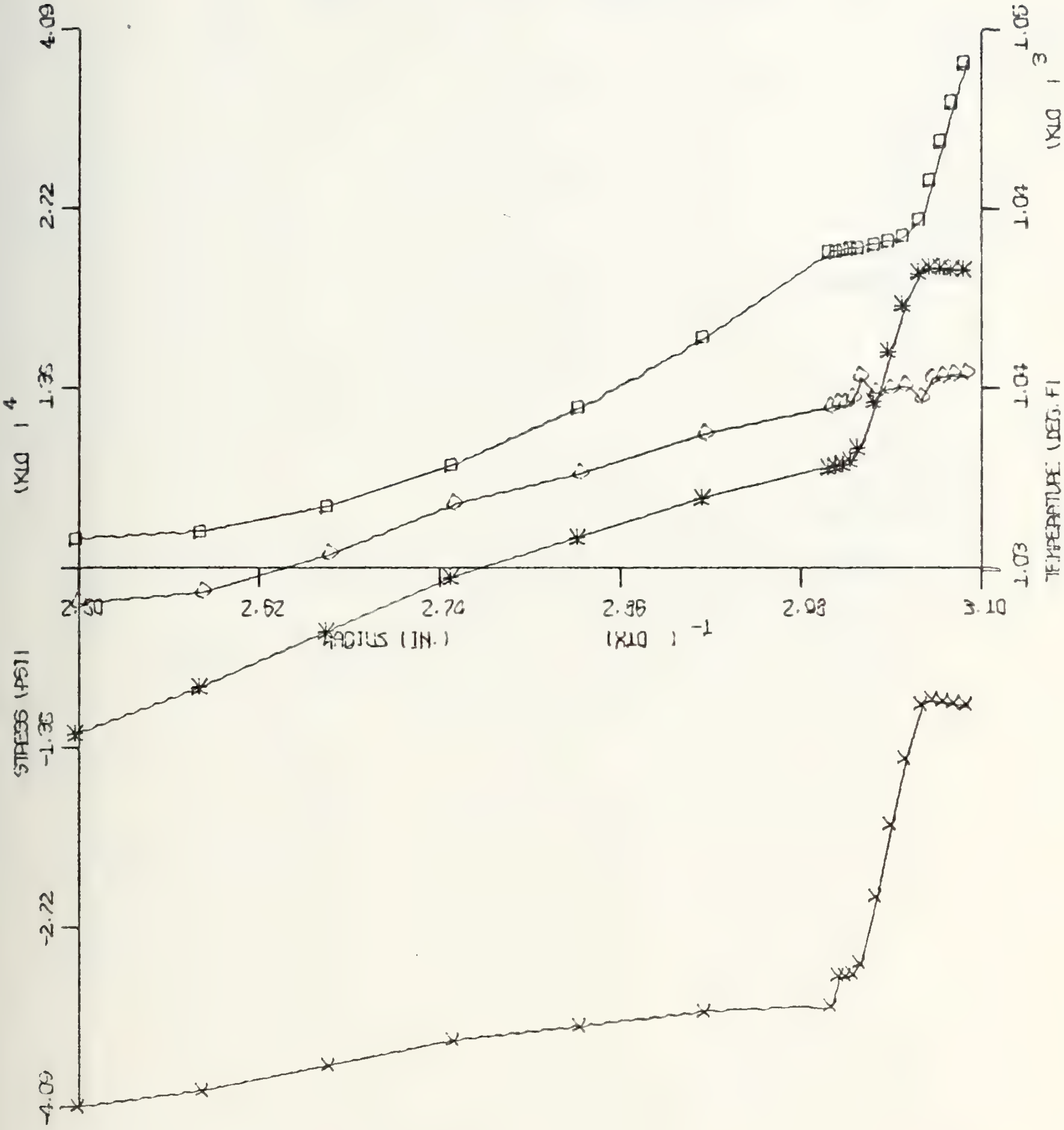

STRESS AND TEMPERATURE
VERSUS RADIAL POSITION
TIME = 0.100 CONF.4-0
SIGMA R - DIAMOND SIGMA T - ASTERISK
SIGMA Z - CROSS TEMPERATURE - SQUARE
STRESS SCALE = 1.722X 10^4 PSI/INCH
TEMPERATURE SCALE = 13.3 DEGREES/INCH
RADIAL POSITION SCALE = 0.01 INCHES/INCH

FIGURE 22(b)

STRESS AND TEMPERATURE
VERSUS RADIAL POSITION
TIME = 0.200 CONF.4-0



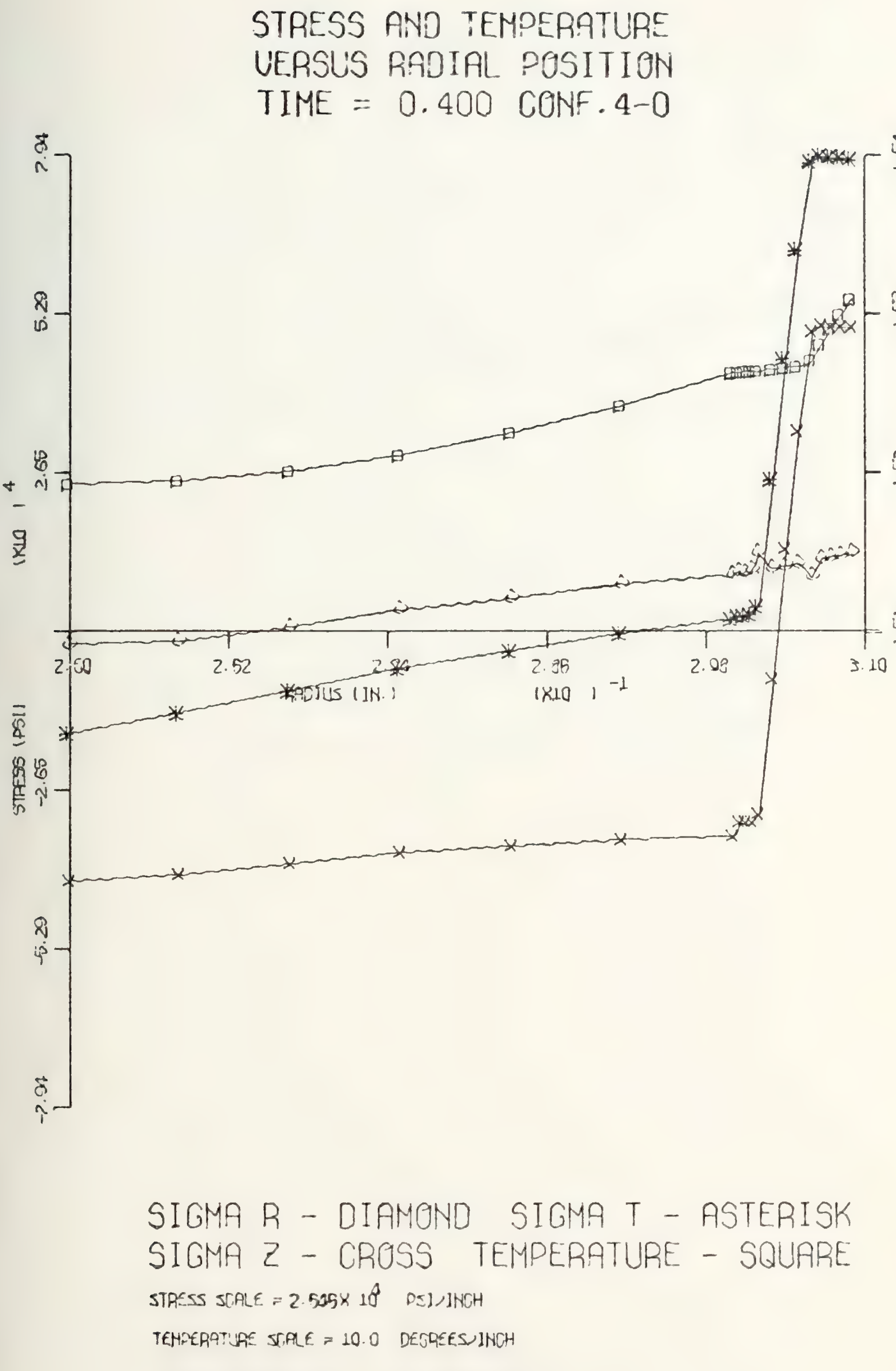
SIGMA R - DIAMOND SIGMA T - ASTERISK
SIGMA Z - CROSS TEMPERATURE - SQUARE

STRESS SCALE = 1.362X 10^3 PSI/INCH

TEMPERATURE SCALE = 6.7 DEGREES/INCH

RADIAL POSITION SCALE = 0.01 INCHES/INCH

FIGURE 22(c)

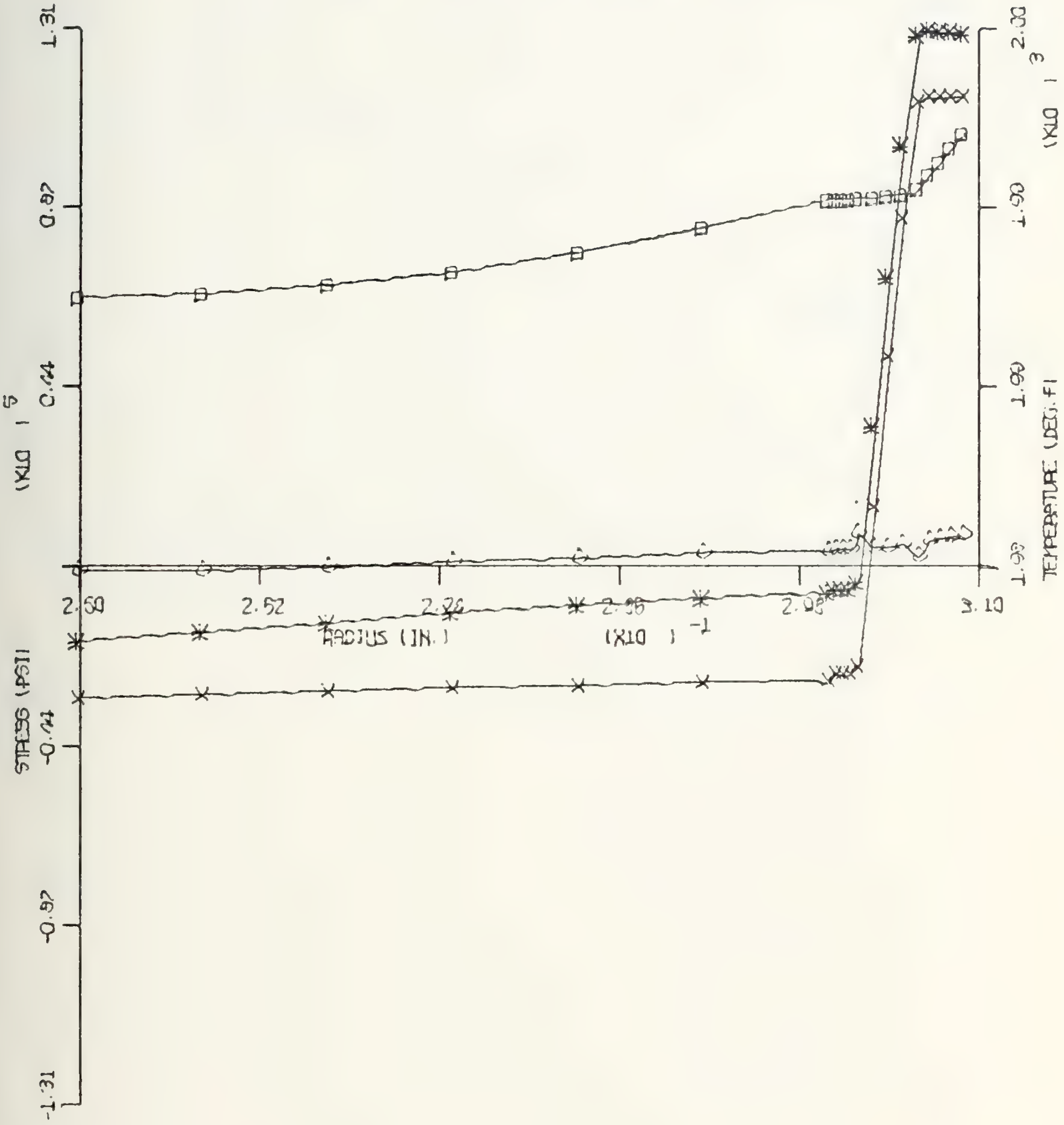


STRESS AND TEMPERATURE
VERSUS RADIAL POSITION
TIME = 0.400 CONF.4-0
2.94
5.29
2.65
-2.65
-5.29
-7.94
(X10) 4
STRESS (PSI)
1.54
1.53
1.52
1.51
(X10) 3
TEMPERATURE (DEG.F)
2.50
2.52
2.94
2.96
2.98
3.10
RADIUS (IN.)
(X10) -1

SIGMA R - DIAMOND SIGMA T - ASTERISK
SIGMA Z - CROSS TEMPERATURE - SQUARE
STRESS SCALE = 2.595X 10 PSI/INCH
TEMPERATURE SCALE = 10.0 DEGREES/INCH
RADIAL POSITION SCALE = 0.01 INCHES/INCH

FIGURE 22(d)

STRESS AND TEMPERATURE
VERSUS RADIAL POSITION
TIME = 0.750 CONF.4-0



SIGMA R - DIAMOND SIGMA T - ASTERISK
SIGMA Z - CROSS TEMPERATURE - SQUARE

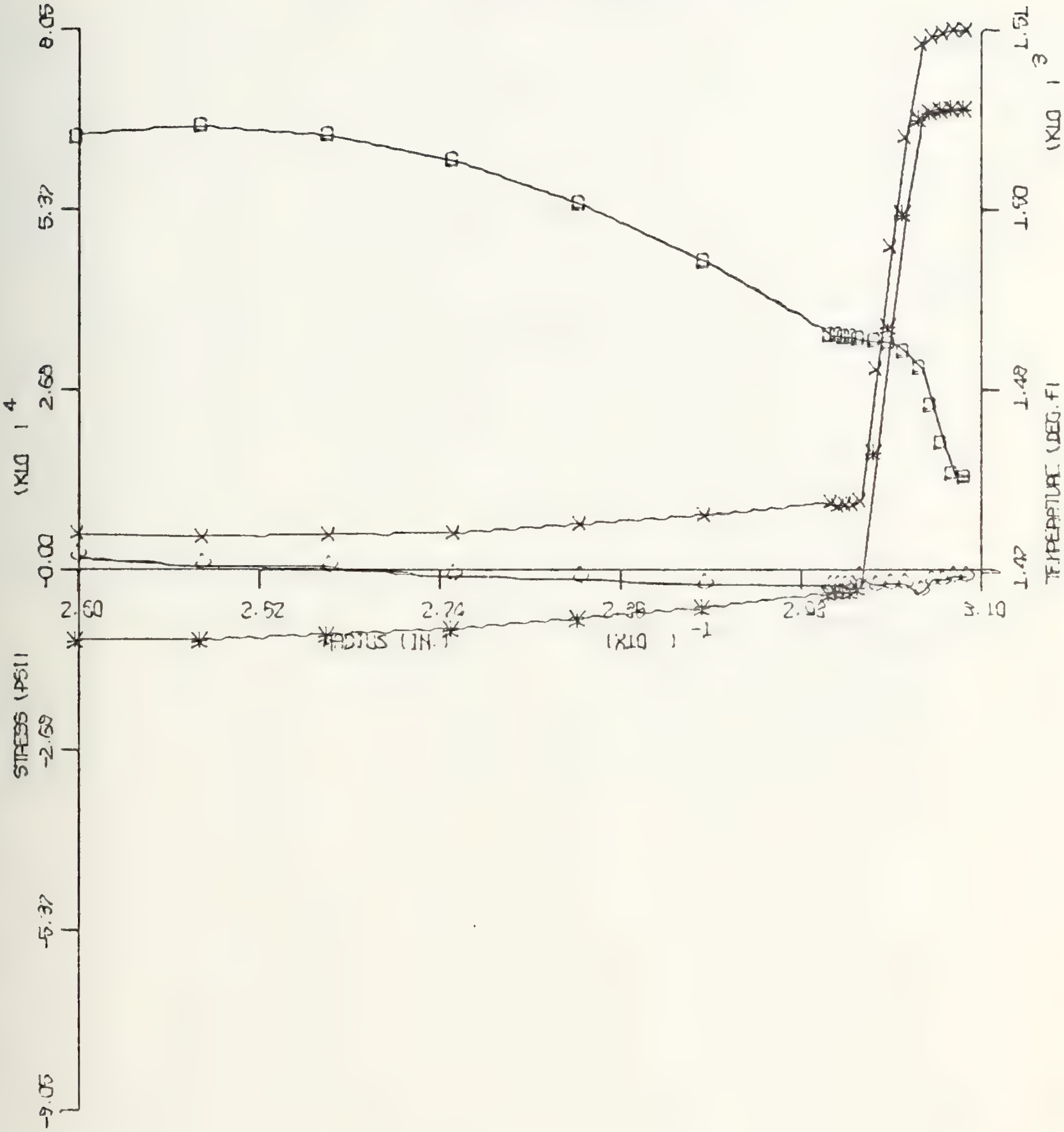
STRESS SCALE = 4.361X 10^8 PSI/INCH

TEMPERATURE SCALE = 5.7 DEGREES/INCH

RADIAL POSITION SCALE = 0.01 INCHES/INCH

FIGURE 22(e)

STRESS AND TEMPERATURE
VERSUS RADIAL POSITION
TIME = 0.800 CONF.4-0



SIGMA R - DIAMOND SIGMA T - ASTERISK
SIGMA Z - CROSS TEMPERATURE - SQUARE

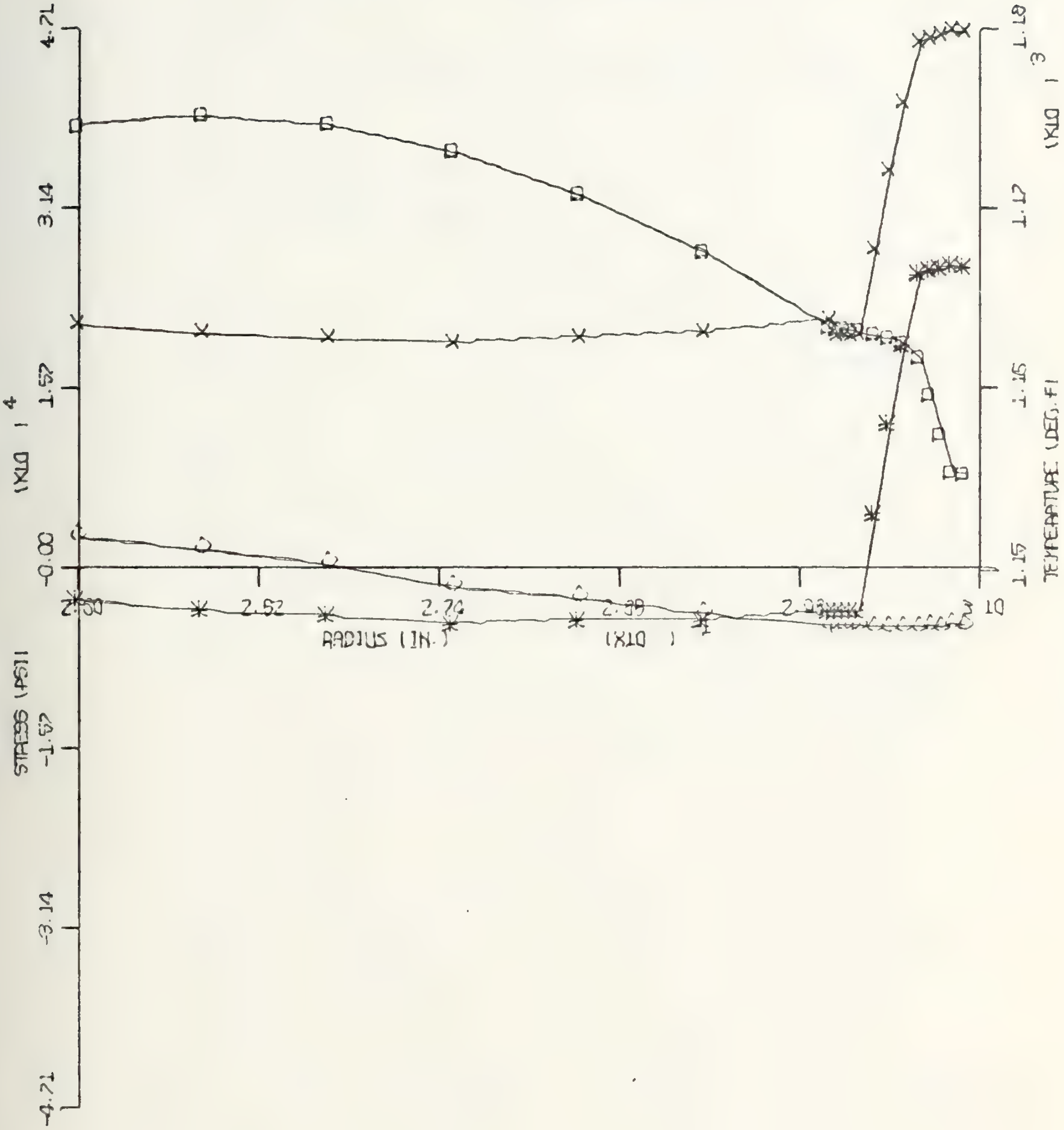
STRESS SCALE = 2.683X 10^4 PSI/INCH

TEMPERATURE SCALE = 13.3 DEGREES/INCH

RADIAL POSITION SCALE = 0.01 INCHES/INCH

FIGURE 22(f)

STRESS AND TEMPERATURE
VERSUS RADIAL POSITION
TIME = 0.850 CONF.4-0



SIGMA R - DIAMOND SIGMA T - ASTERISK
SIGMA Z - CROSS TEMPERATURE - SQUARE

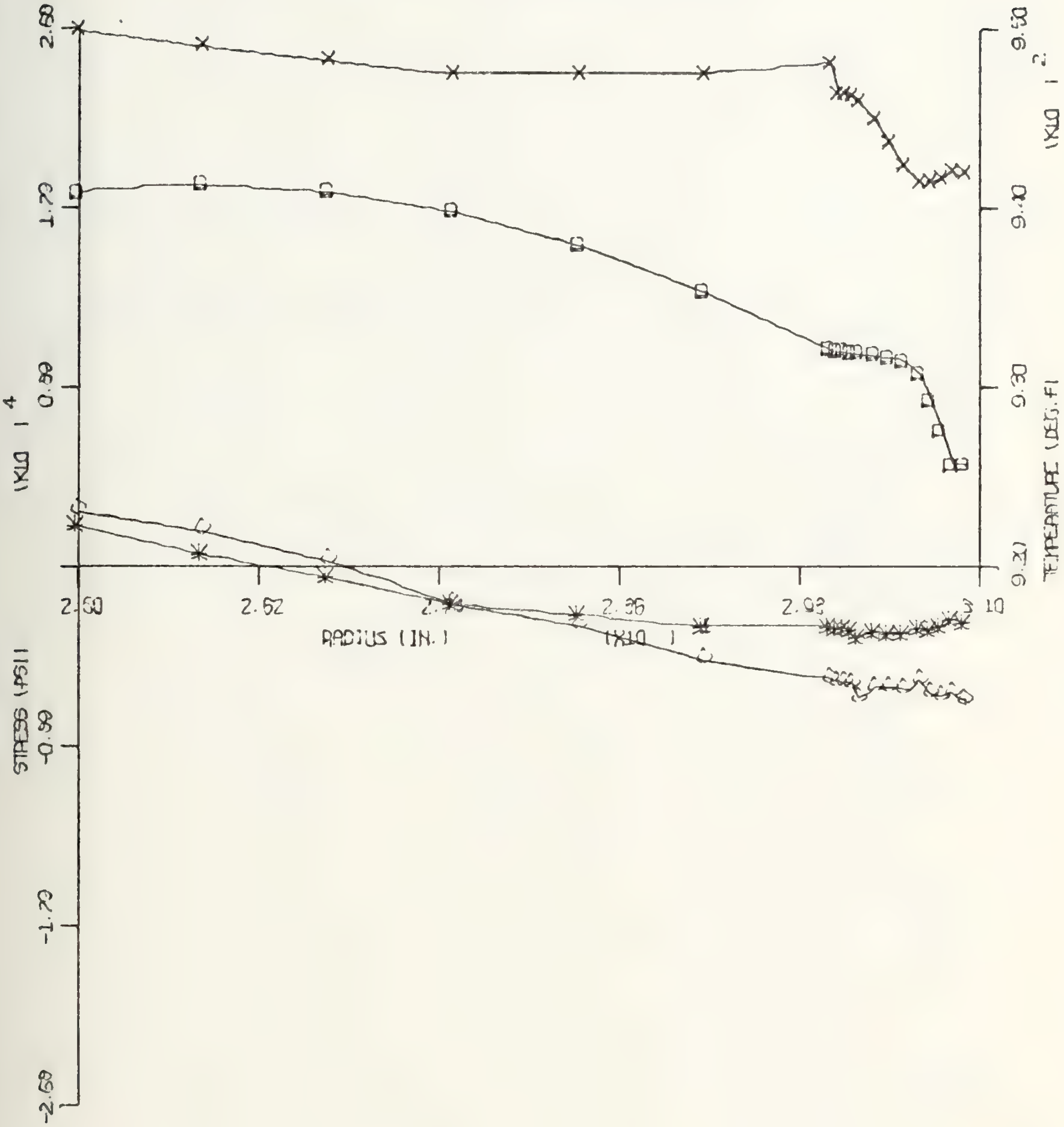
STRESS SCALE = 1.570X 10^4 PSI/INCH

TEMPERATURE SCALE = 10.0 DEGREES/INCH

RADIAL POSITION SCALE = 0.01 INCHES/INCH

FIGURE 22(g)

STRESS AND TEMPERATURE
VERSUS RADIAL POSITION
TIME = 0.900 CONF.4-0



SIGMA R - DIAMOND SIGMA T - ASTERISK
SIGMA Z - CROSS TEMPERATURE - SQUARE

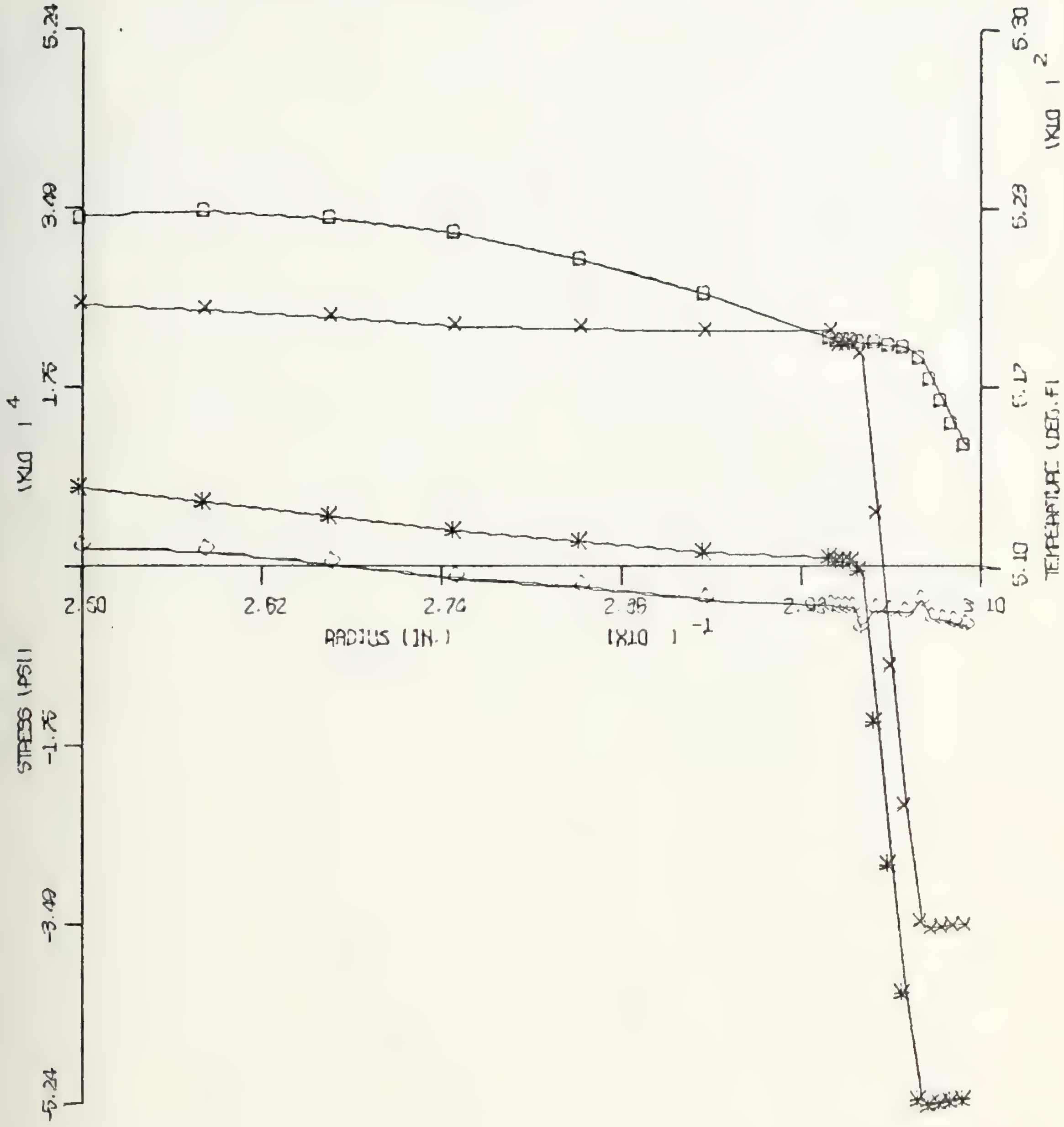
STRESS SCALE = 3.033X 10^3 PSI/INCH

TEMPERATURE SCALE = 10.0 DEGREES/INCH

RADIAL POSITION SCALE = 0.01 INCHES/INCH

FIGURE 22(h)

STRESS AND TEMPERATURE
VERSUS RADIAL POSITION
TIME = 1.050 CONF.4-0



SIGMA R - DIAMOND SIGMA T - ASTERISK
SIGMA Z - CROSS TEMPERATURE - SQUARE

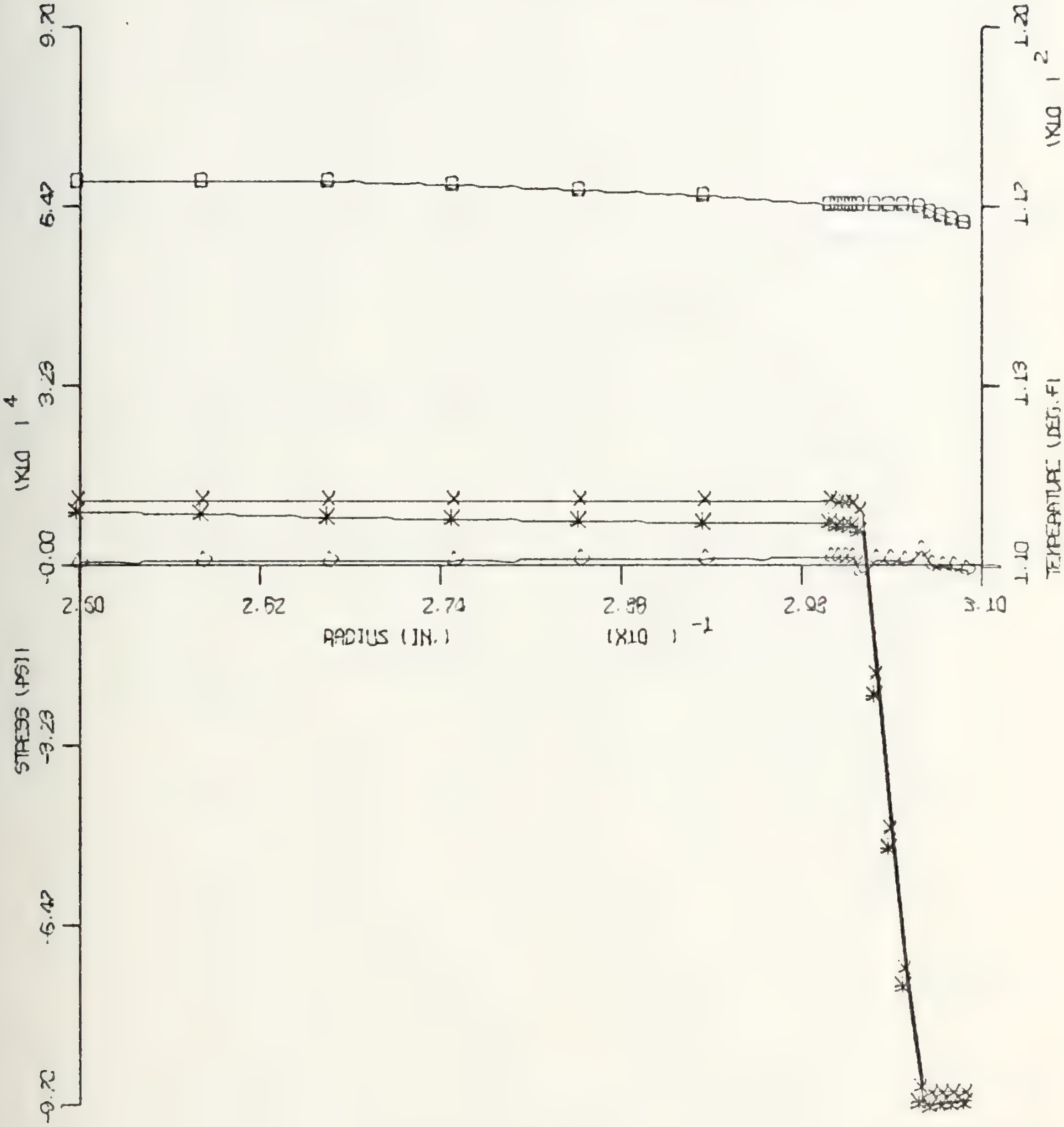
STRESS SCALE = 1.747X 10^4 PSI/INCH

TEMPERATURE SCALE = 5.7 DEGREES/INCH

RADIAL POSITION SCALE = 0.01 INCHES/INCH

FIGURE 22(i)

STRESS AND TEMPERATURE
VERSUS RADIAL POSITION
TIME = 1.650 CONF.4-0



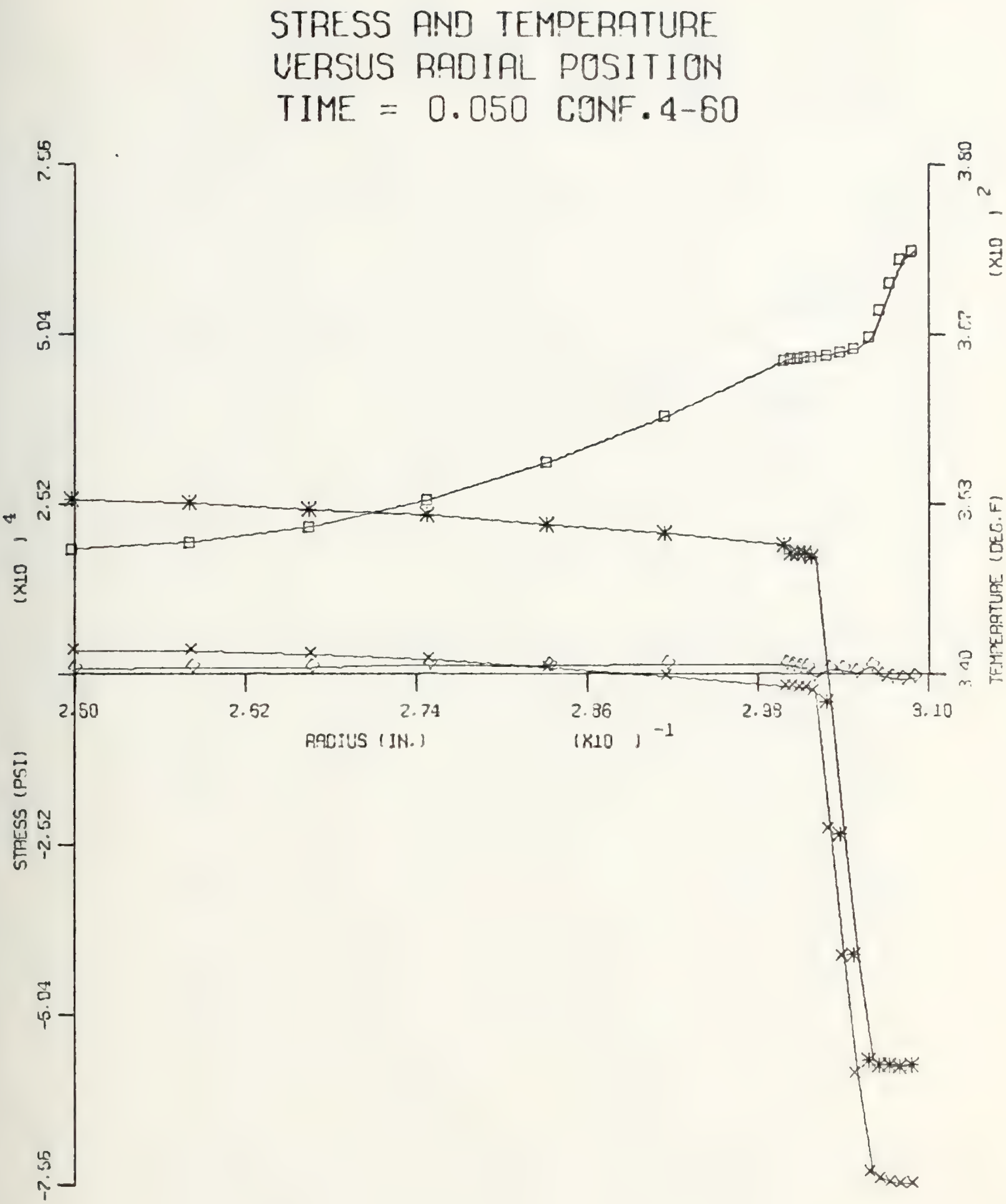
SIGMA R - DIAMOND SIGMA T - ASTERISK
SIGMA Z - CROSS TEMPERATURE - SQUARE

STRESS SCALE = 3.233X 10^{3} PSI/INCH

TEMPERATURE SCALE = 3.3 DEGREES/INCH

RADIAL POSITION SCALE = 0.01 INCHES/INCH

FIGURE 22(j)



SIGMA R - DIAMOND SIGMA T - ASTERISK
SIGMA Z - CROSS TEMPERATURE - SQUARE

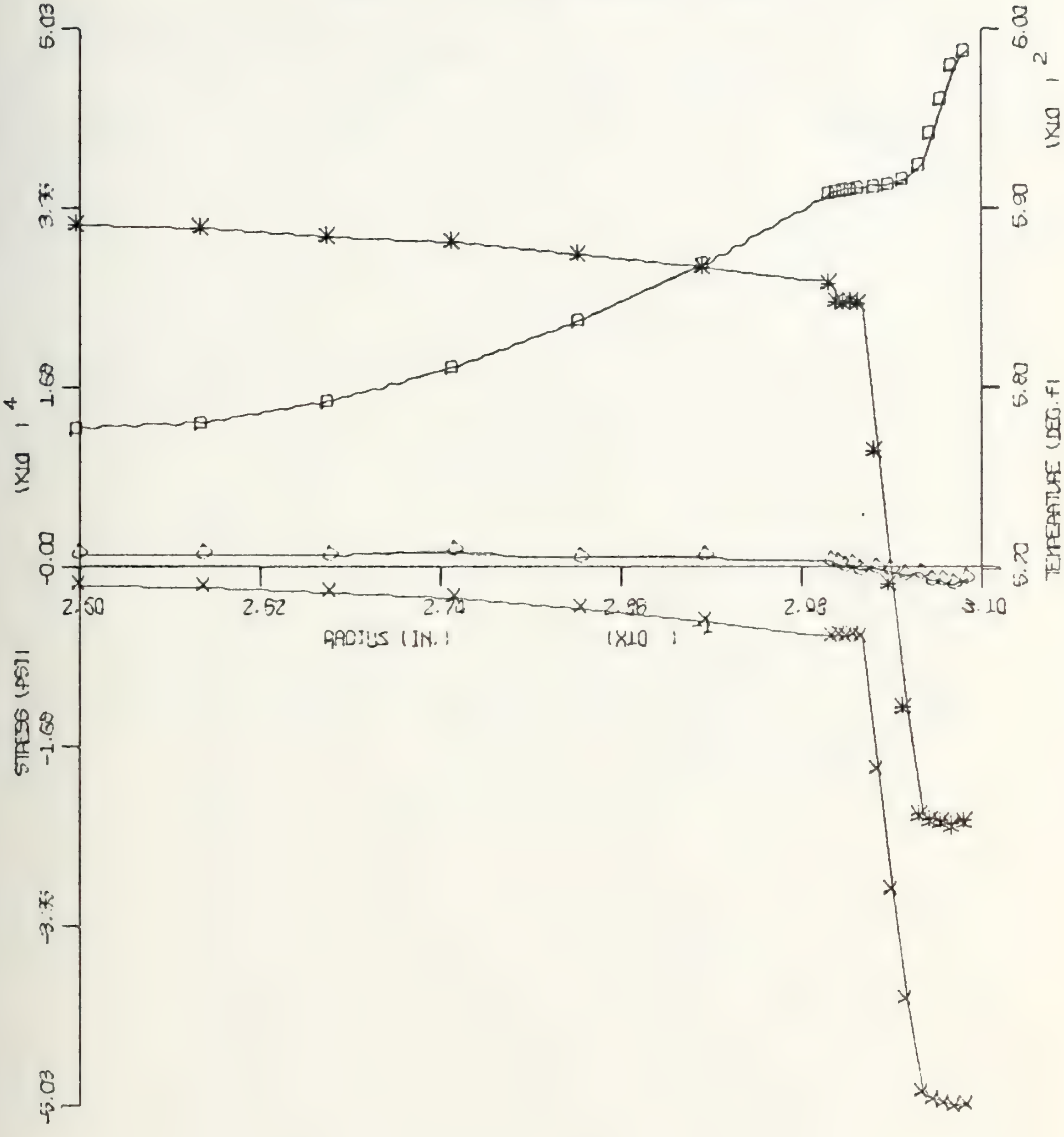
STRESS SCALE = 2.518×10^4 PSI/INCH

TEMPERATURE SCALE = 13.3 DEGREES/INCH

RADIAL POSITION SCALE = 0.01 INCHES/INCH

FIGURE 23(a)

STRESS AND TEMPERATURE
VERSUS RADIAL POSITION
TIME = 0.100 CONF.4-60



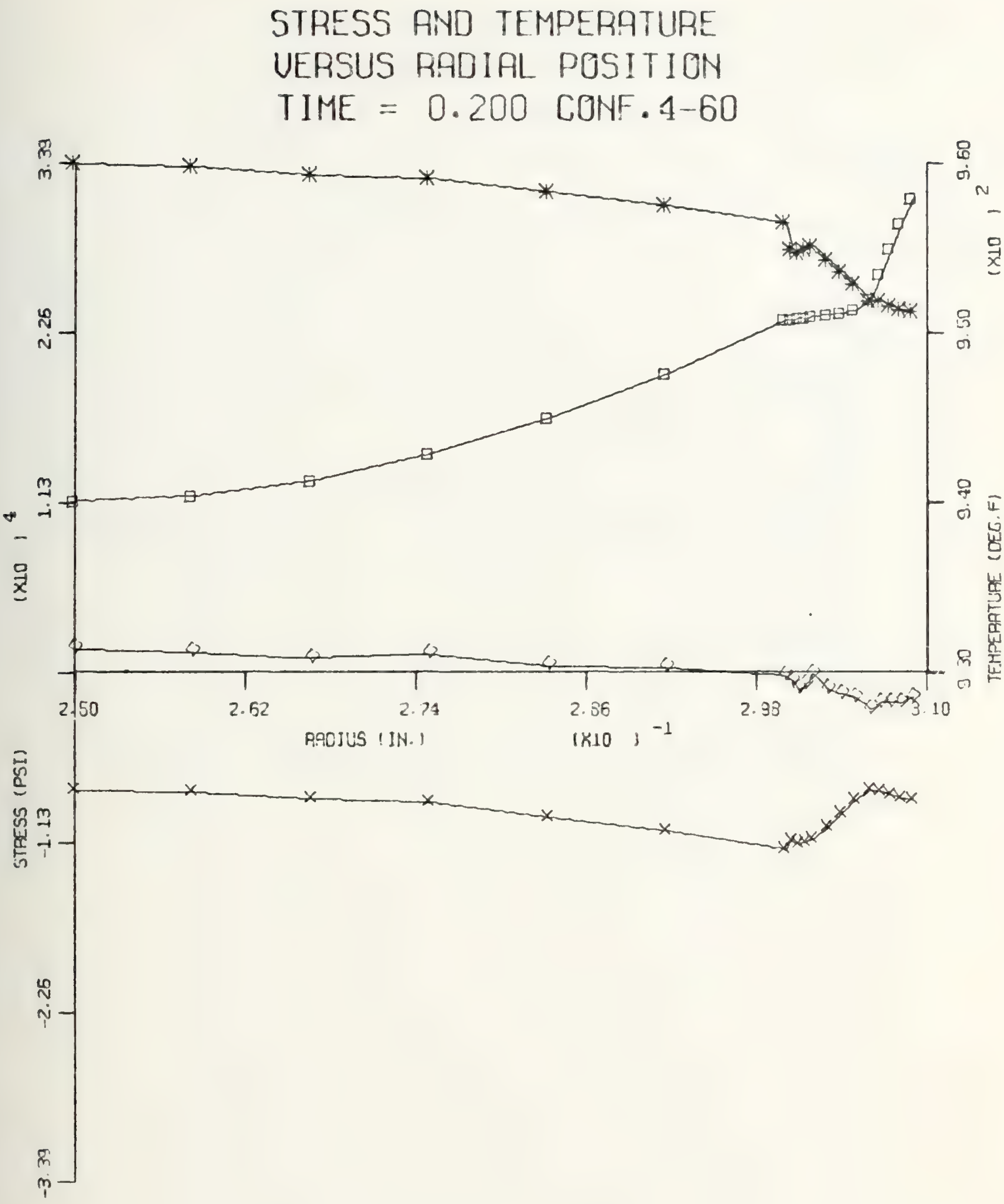
SIGMA R - DIAMOND SIGMA T - ASTERISK
SIGMA Z - CROSS TEMPERATURE - SQUARE

STRESS SCALE = 1.576X 10^{4} PSI/INCH

TEMPERATURE SCALE = 10.0 DEGREES/INCH

RADIAL POSITION SCALE = 0.01 INCHES/INCH

FIGURE 23(b)



SIGMA R - DIAMOND SIGMA T - ASTERISK
SIGMA Z - CROSS TEMPERATURE - SQUARE

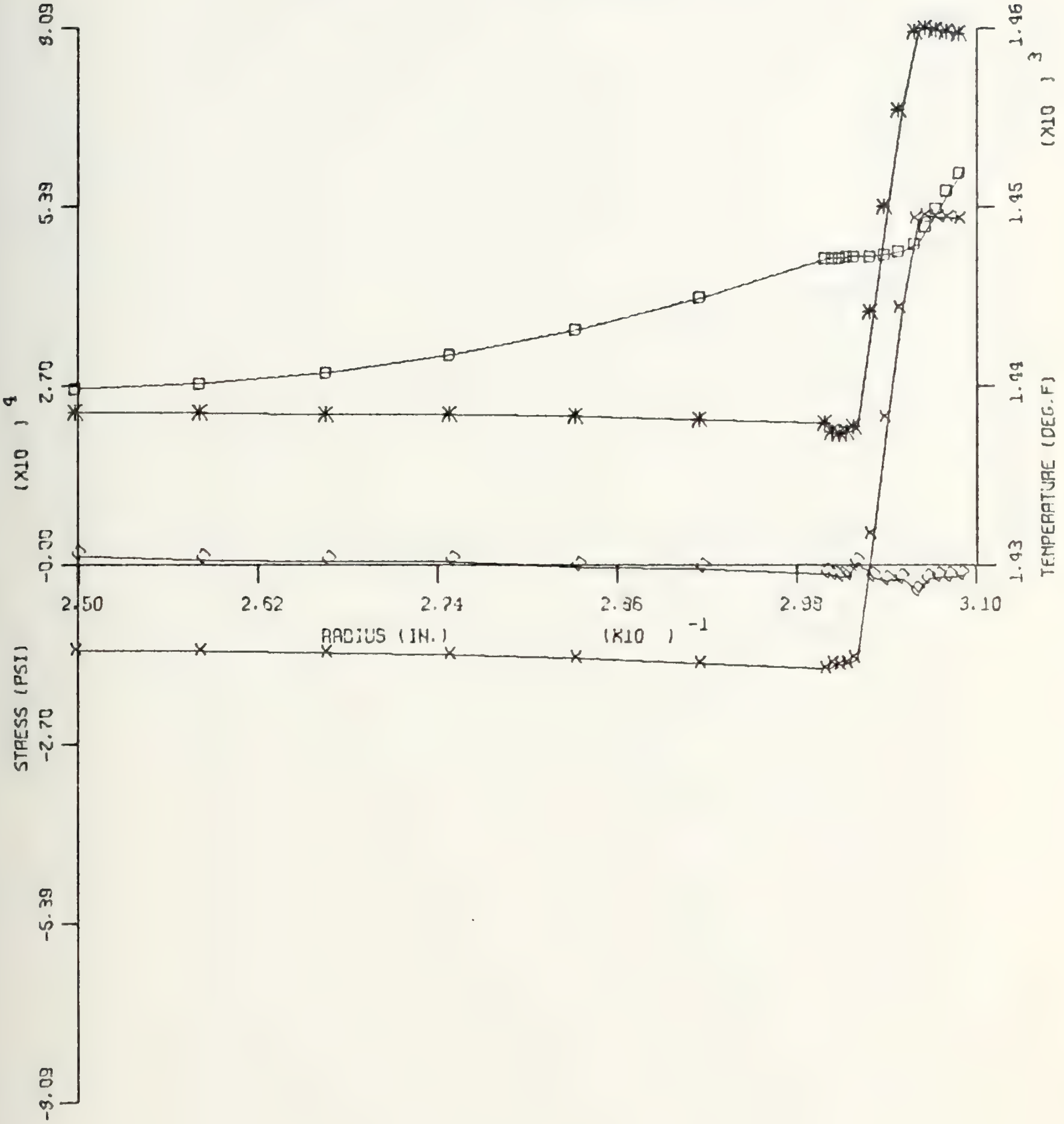
STRESS SCALE = 1.130×10^4 PSI/INCH

TEMPERATURE SCALE = 10.0 DEGREES/INCH

RADIAL POSITION SCALE = 0.01 INCHES/INCH

FIGURE 23(c)

STRESS AND TEMPERATURE
VERSUS RADIAL POSITION
TIME = 0.400 CONF.4-60



SIGMA R - DIAMOND SIGMA T - ASTERISK
SIGMA Z - CROSS TEMPERATURE - SQUARE

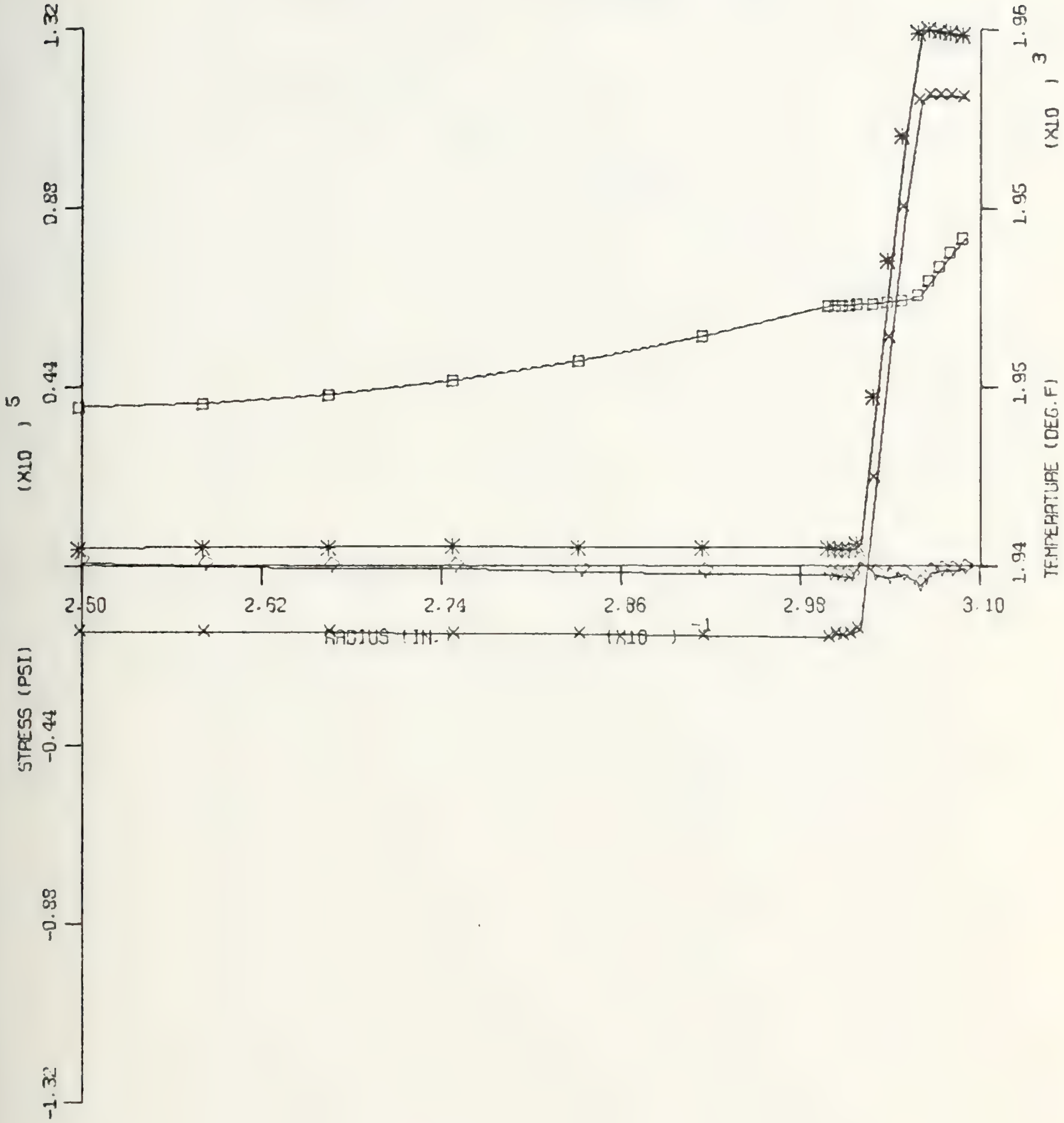
STRESS SCALE = 2.697X 10^4 PSI/INCH

TEMPERATURE SCALE = 10.0 DEGREES/INCH

RADIAL POSITION SCALE = 0.01 INCHES/INCH

FIGURE 23(d)

STRESS AND TEMPERATURE
VERSUS RADIAL POSITION
TIME = 0.750 CONF.4-60



SIGMA R - DIAMOND SIGMA T - ASTERISK
SIGMA Z - CROSS TEMPERATURE - SQUARE

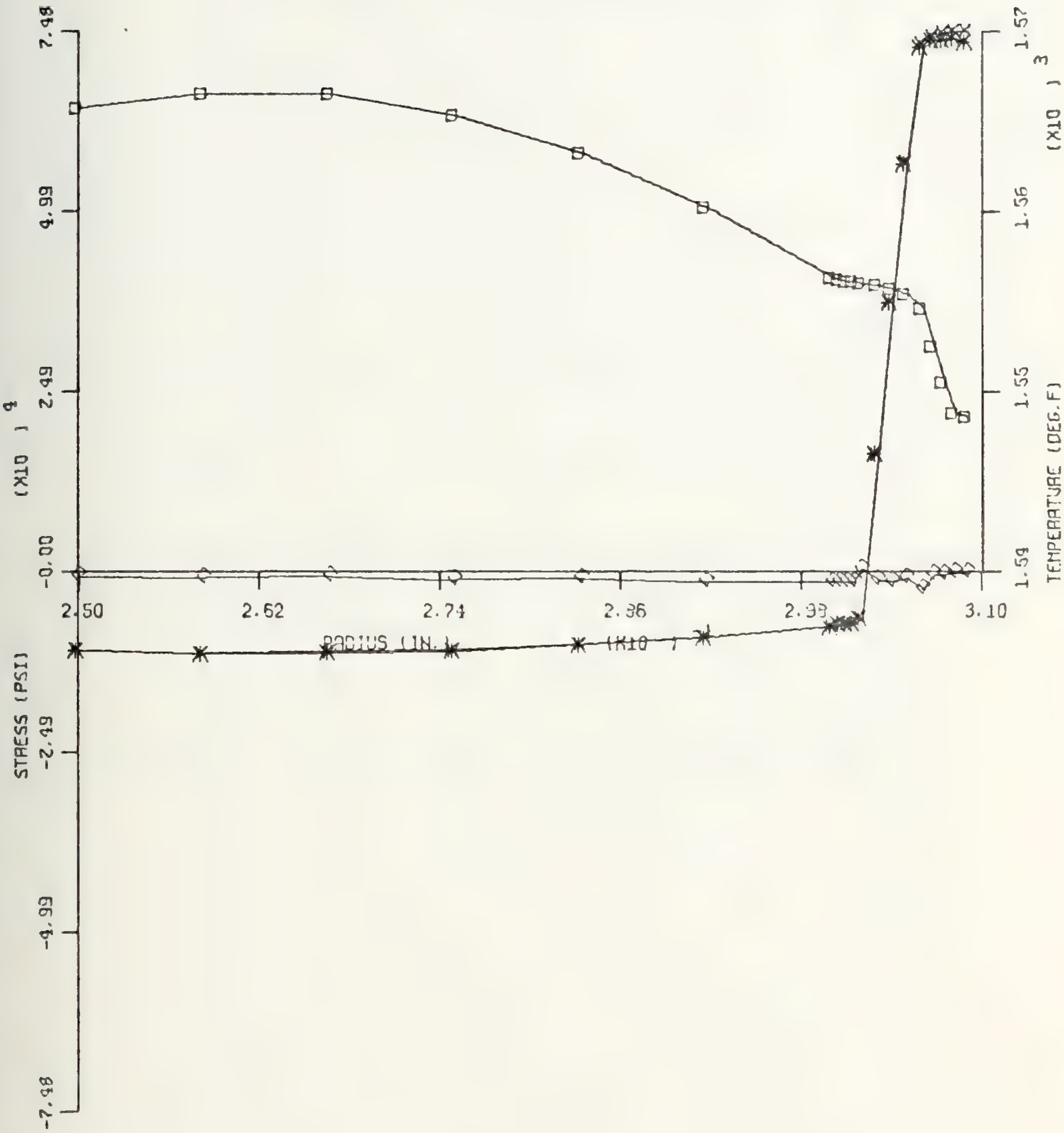
STRESS SCALE = 1.369X 10^3 PSI/INCH

TEMPERATURE SCALE = 6.7 DEGREES/INCH

RADIAL POSITION SCALE = 0.01 INCHES/INCH

FIGURE 23(e)

STRESS AND TEMPERATURE
VERSUS RADIAL POSITION
TIME = 0.800 CONF.4-60



SIGMA R - DIAMOND SIGMA T - ASTERISK
SIGMA Z - CROSS TEMPERATURE - SQUARE

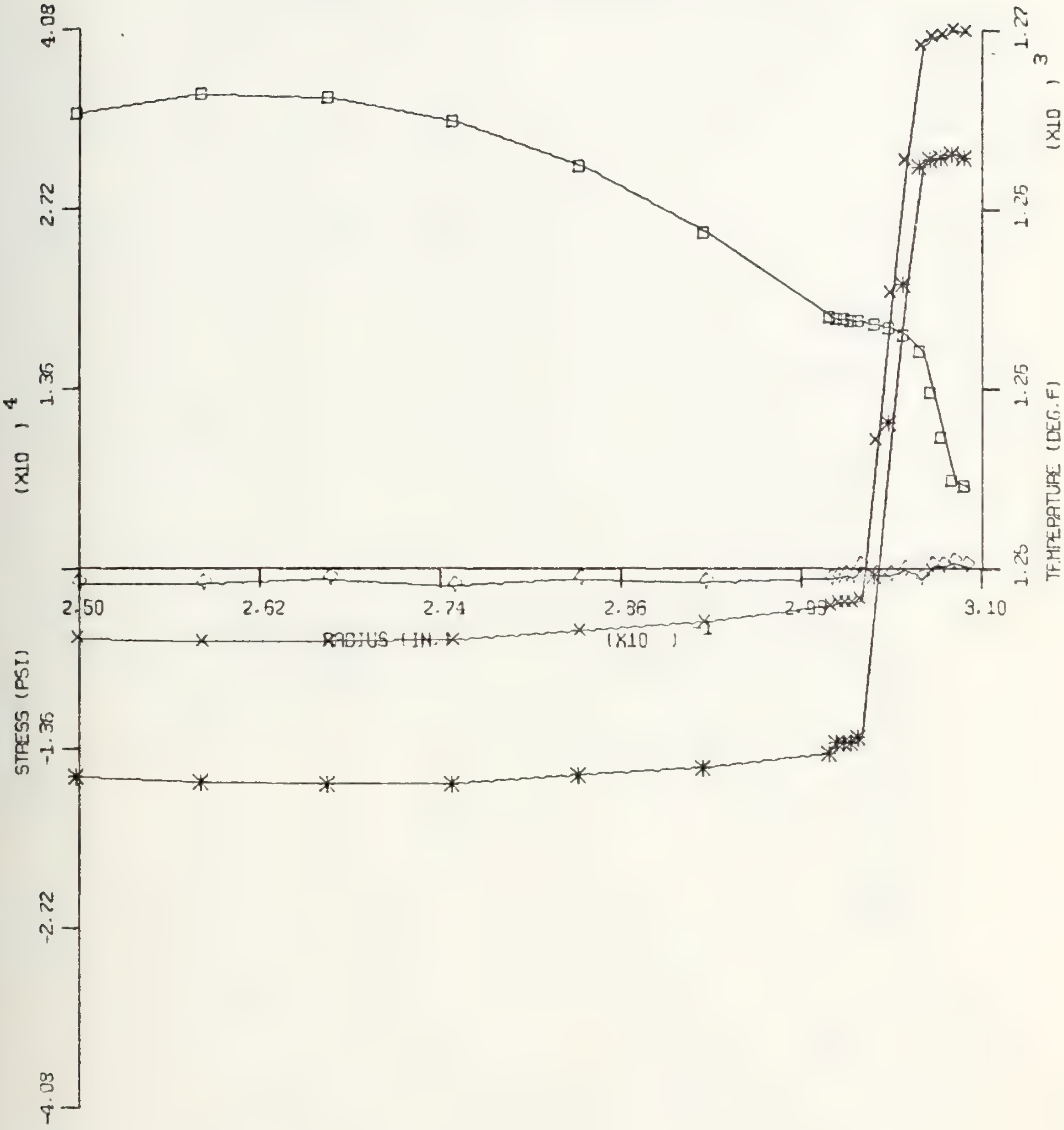
STRESS SCALE = 2.493K 10^4 PSI/INCH

TEMPERATURE SCALE = 10.0 DEGREES/INCH

RADIAL POSITION SCALE = 0.01 INCHES/INCH

FIGURE 23(f)

STRESS AND TEMPERATURE
VERSUS RADIAL POSITION
TIME = 0.850 CONF.4-60



SIGMA R - DIAMOND SIGMA T - ASTERISK
SIGMA Z - CROSS TEMPERATURE - SQUARE

STRESS SCALE = 1.358X 10^4 PSI/INCH

TEMPERATURE SCALE = 6.7 DEGREES/INCH

RADIAL POSITION SCALE = 0.01 INCHES/INCH

FIGURE 23(g)

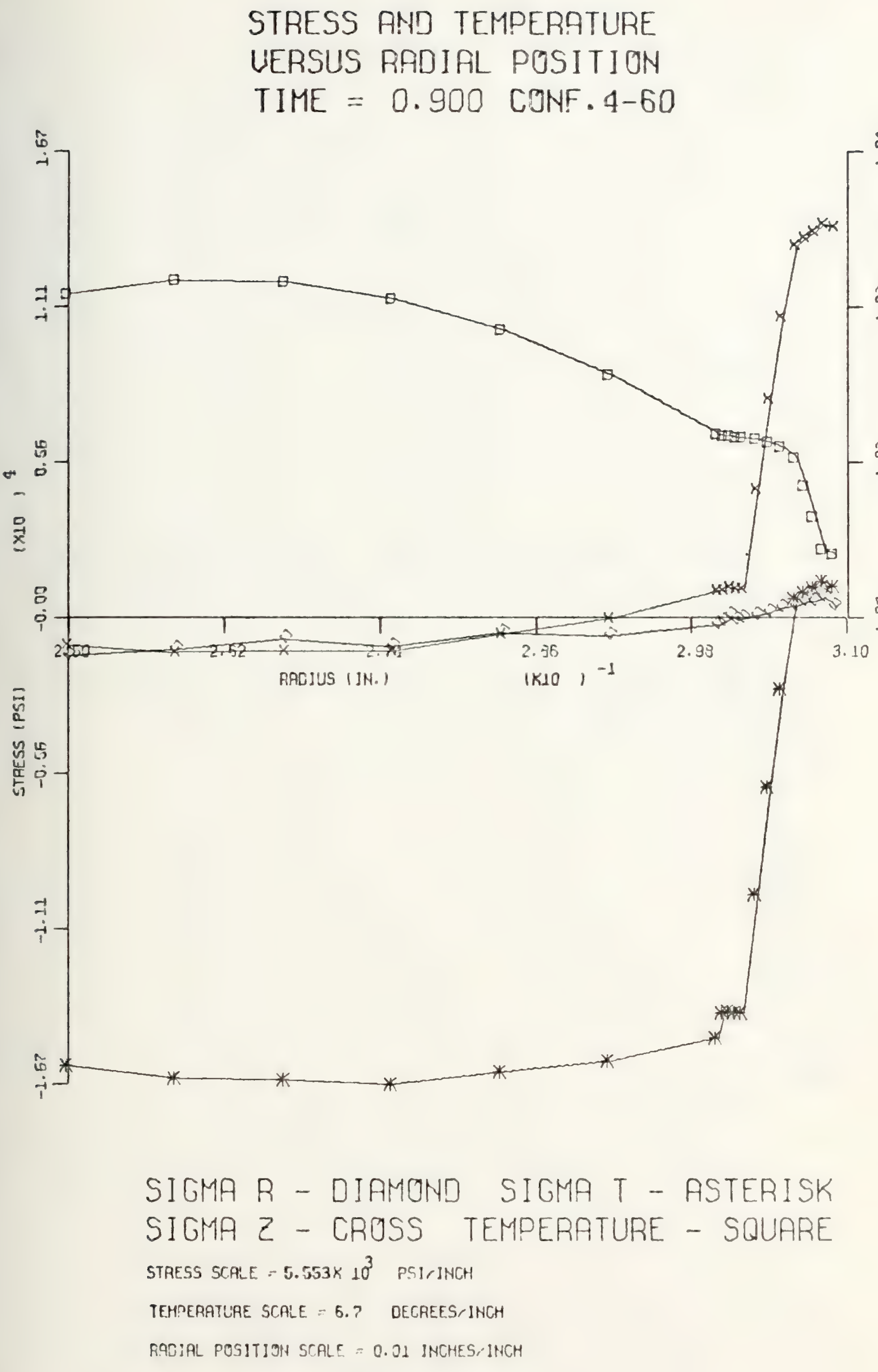
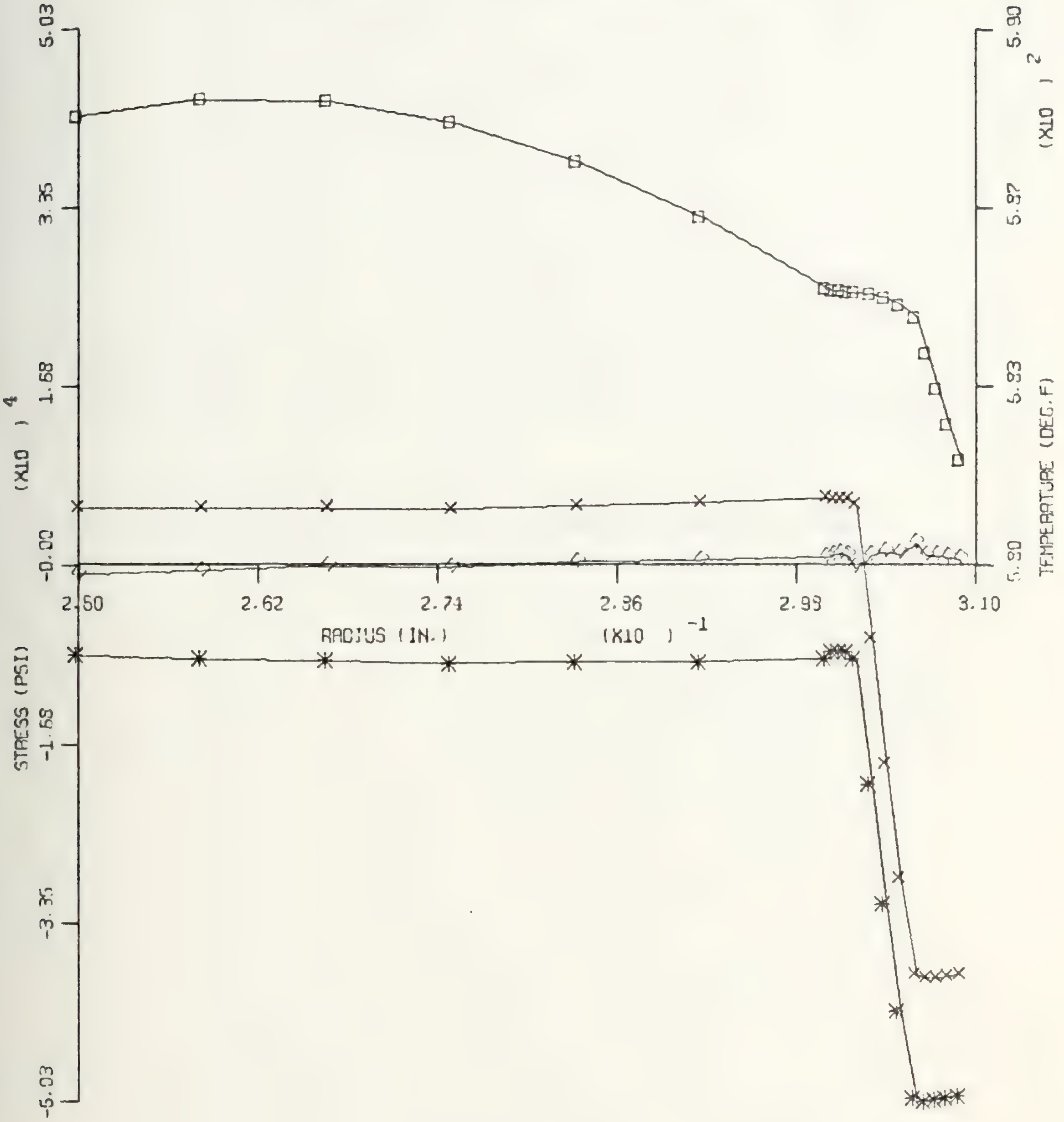

STRESS AND TEMPERATURE
VERSUS RADIAL POSITION
TIME = 0.900 CONF.4-60
STRESS (PSI)
(X10) 4
TEMPERATURE (DEG.F)
(X10) 3
RADIUS (IN.)
(X10) -1
SIGMA R - DIAMOND SIGMA T - ASTERISK
SIGMA Z - CROSS TEMPERATURE - SQUARE
STRESS SCALE = 5.553X 10 3 PSI/INCH
TEMPERATURE SCALE = 6.7 DEGREES/INCH
RADIAL POSITION SCALE = 0.01 INCHES/INCH

FIGURE 23(h)

STRESS AND TEMPERATURE
VERSUS RADIAL POSITION
TIME = 1.050 CONF.4-60



SIGMA R - DIAMOND SIGMA T - ASTERISK
SIGMA Z - CROSS TEMPERATURE - SQUARE

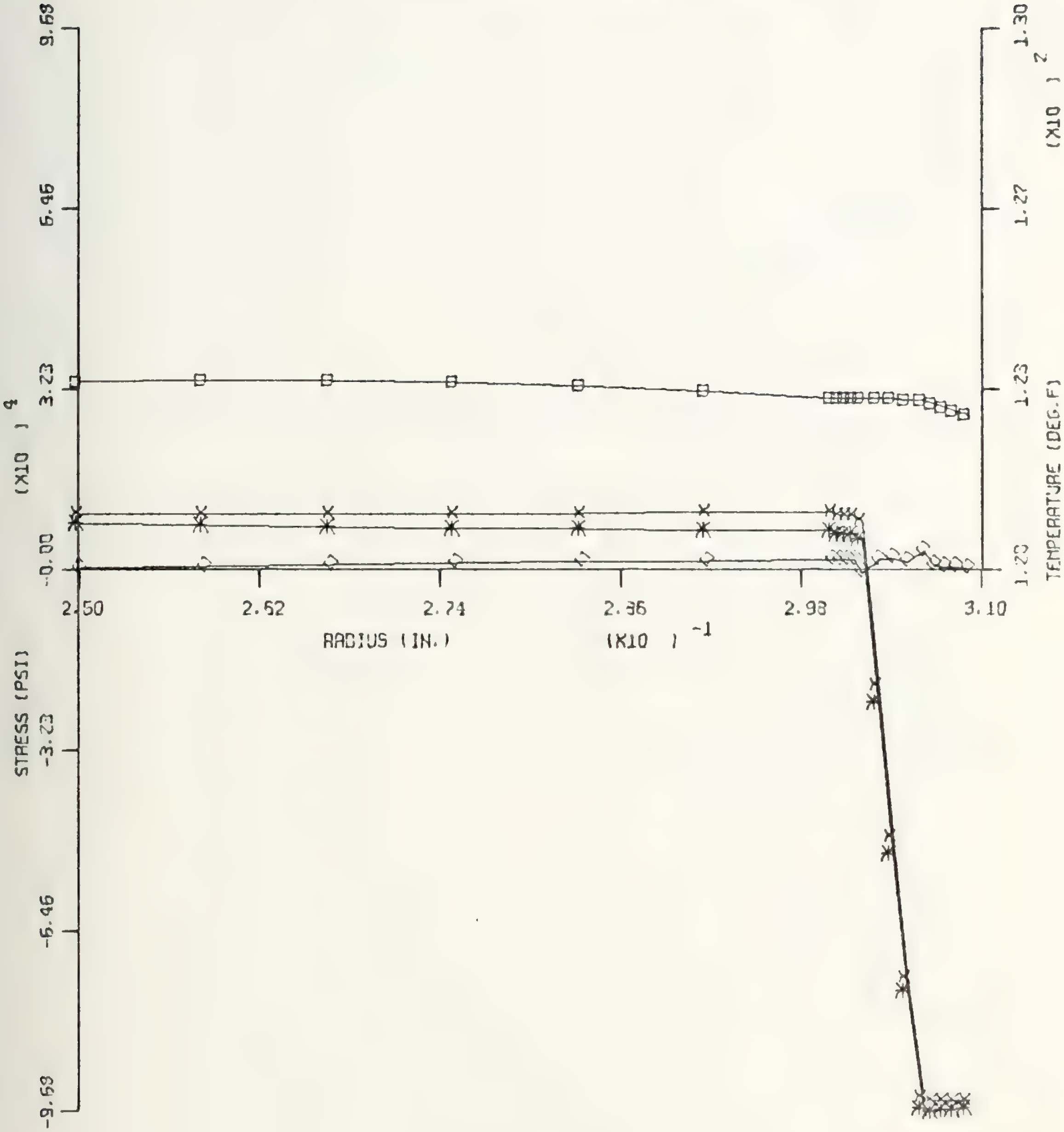
STRESS SCALE = 1.675×10^4 PSI/INCH

TEMPERATURE SCALE = 3.3 DEGREES/INCH

RADIAL POSITION SCALE = 0.01 INCHES/INCH

FIGURE 23(i)

STRESS AND TEMPERATURE
VERSUS RADIAL POSITION
TIME = 1.650 CONF.4-60



SIGMA R - DIAMOND SIGMA T - ASTERISK
SIGMA Z - CROSS TEMPERATURE - SQUARE

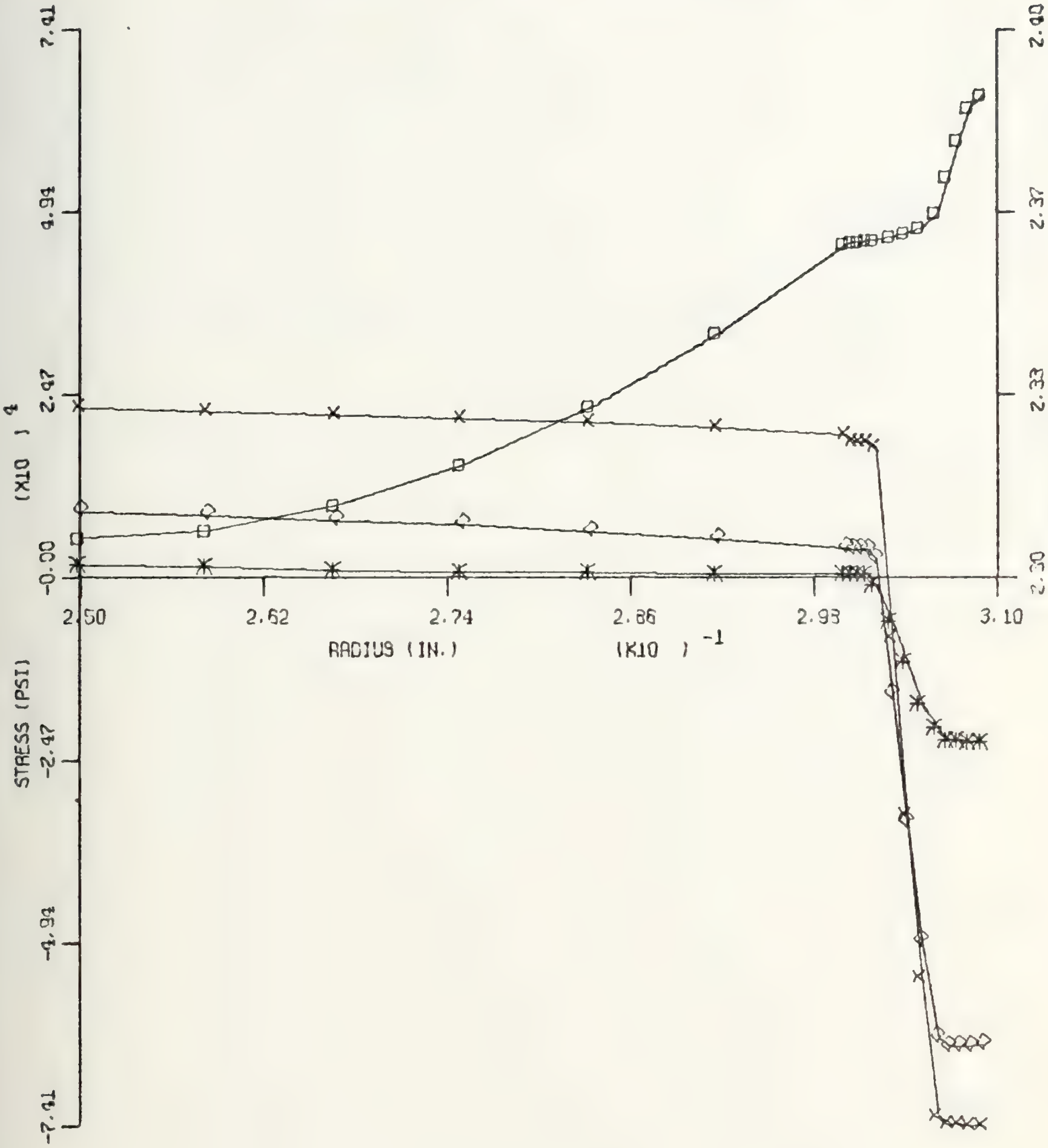
STRESS SCALE = 3.228X 10^4 PSI/INCH

TEMPERATURE SCALE = 3.3 DEGREES/INCH

RADIAL POSITION SCALE = 0.01 INCHES/INCH

FIGURE 23(j)

STRESS AND TEMPERATURE
VERSUS RADIAL POSITION
TIME = 0.050 CONF.4-120



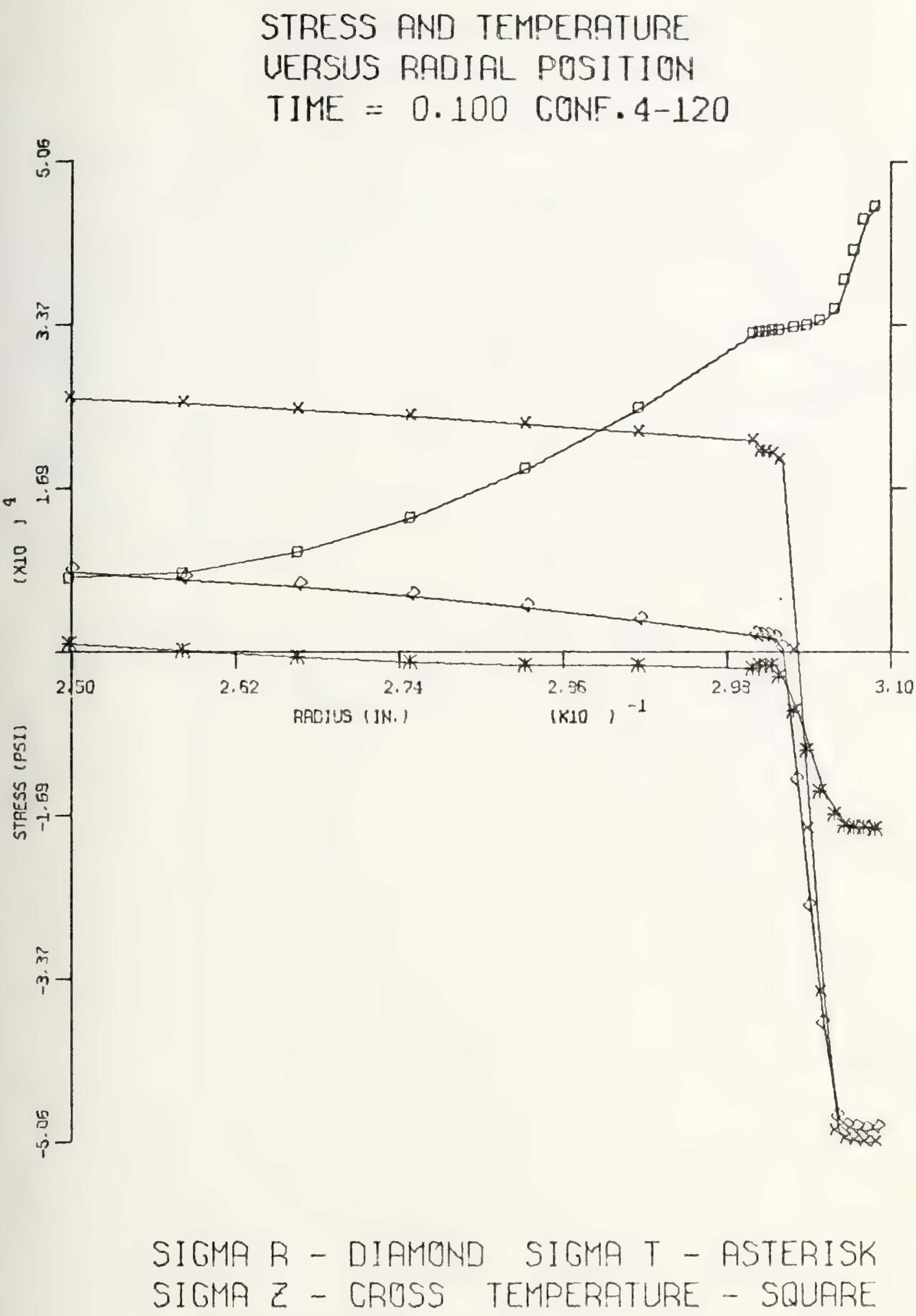
SIGMA R - DIAMOND SIGMA T - ASTERISK
SIGMA Z - CROSS TEMPERATURE - SQUARE

STRESS SCALE = 2.470X 10^4 PSI/INCH

TEMPERATURE SCALE = 3.3 DEGREES/INCH

RADIAL POSITION SCALE = 0.01 INCHES/INCH

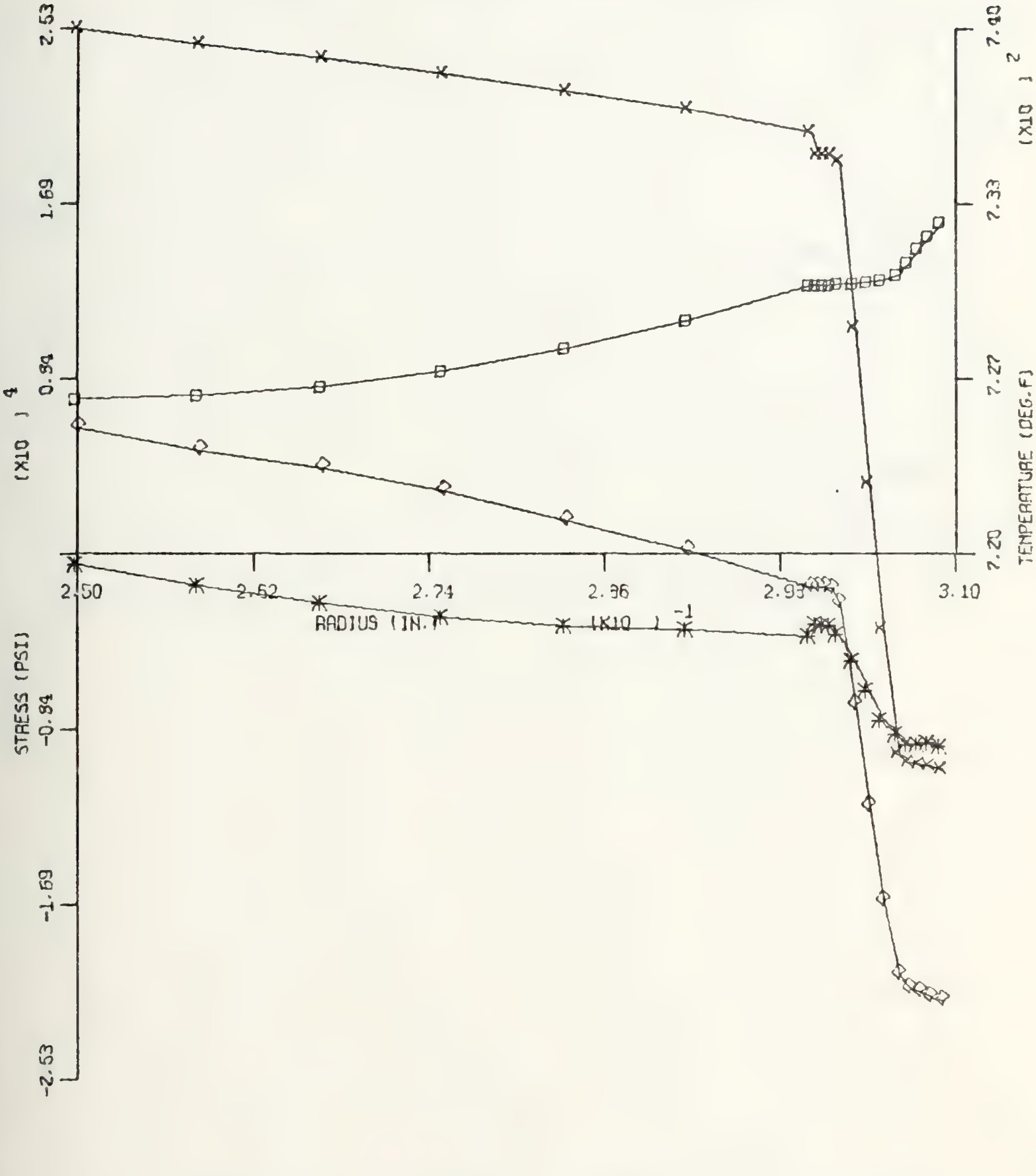
FIGURE 24(a)



STRESS AND TEMPERATURE
VERSUS RADIAL POSITION
TIME = 0.100 CONF.4-120
5.06
3.37
1.69
-1.69
-3.37
-5.06
STRESS (PSI)
(X10)4
2.50
2.62
2.74
2.86
2.98
3.10
RADIUS (IN.)
(X10)-1
TEMPERATURE (DEG.F)
(X10)2
SIGMA R - DIAMOND SIGMA T - ASTERISK
SIGMA Z - CROSS TEMPERATURE - SQUARE
STRESS SCALE = 1.687K 10⁴ PSI/INCH
TEMPERATURE SCALE = 3.3 DEGREES/INCH
RADIAL POSITION SCALE = 0.01 INCHES/INCH

FIGURE 24(b)

STRESS AND TEMPERATURE
VERSUS RADIAL POSITION
TIME = 0.200 CONF.4-120



SIGMA R - DIAMOND SIGMA T - ASTERISK
SIGMA Z - CROSS TEMPERATURE - SQUARE

STRESS SCALE = 8.447X 10^3 PSI/INCH

TEMPERATURE SCALE = 6.7 DEGREES/INCH

RADIAL POSITION SCALE = 0.01 INCHES/INCH

FIGURE 24(c)

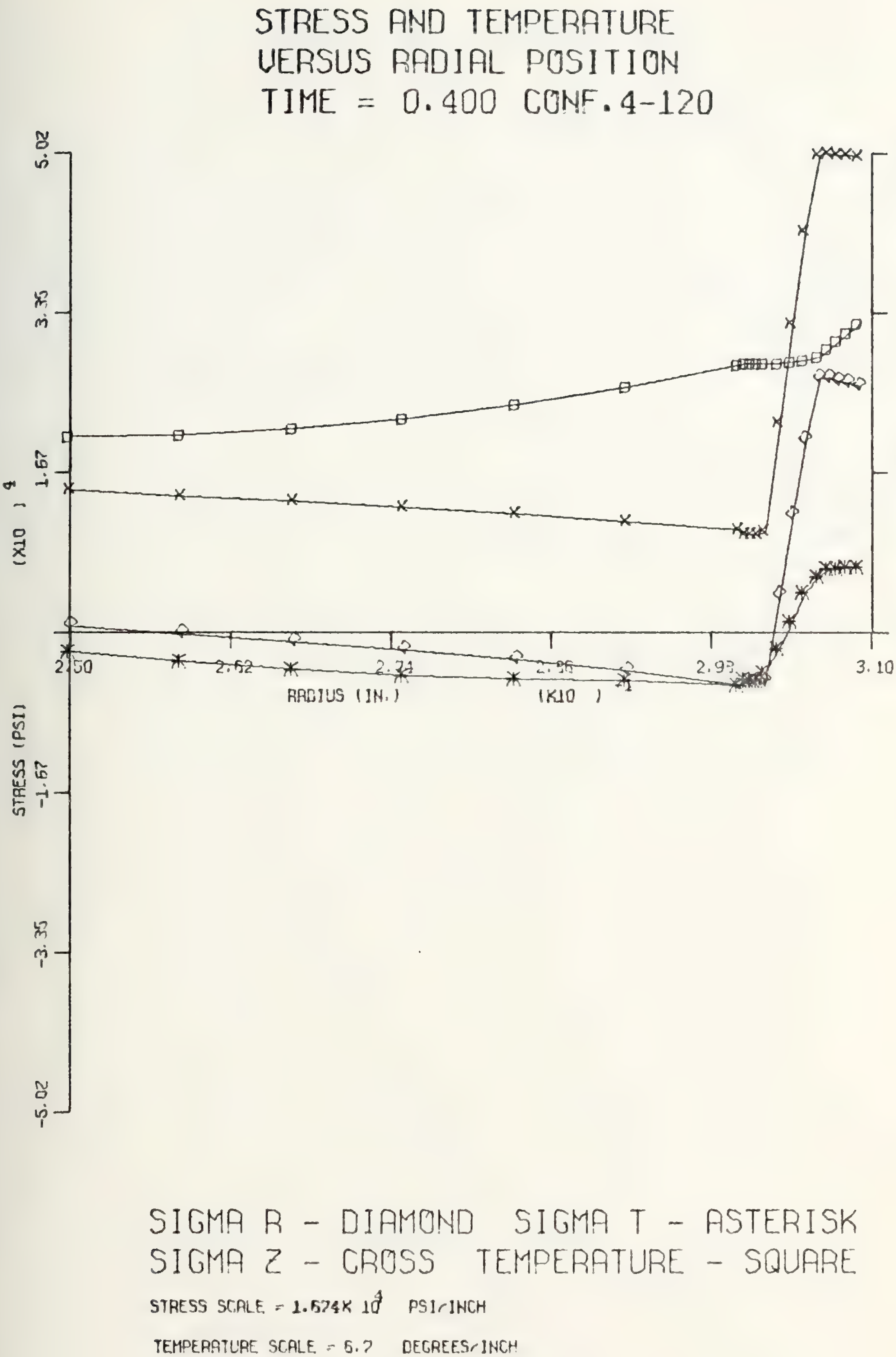
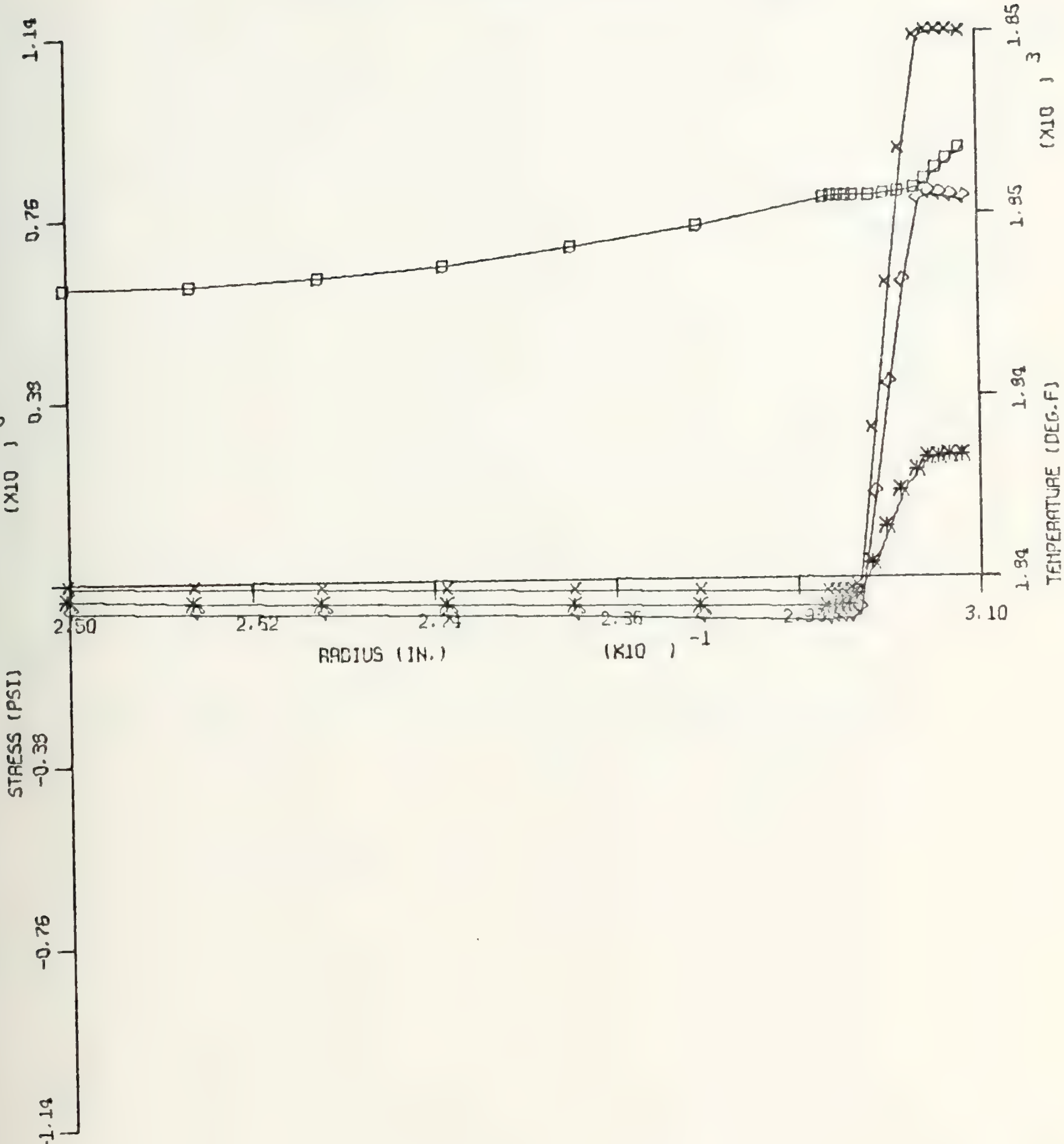

STRESS AND TEMPERATURE
VERSUS RADIAL POSITION
TIME = 0.400 CONF.4-120
STRESS (PSI)
(X10) 4
5.02
3.35
1.67
-1.67
-3.35
-5.02
TEMPERATURE (DEG.F)
(X10) 3
1.27
1.26
1.26
1.25
RADIUS (IN.)
(X10) -1
2.50
2.62
2.74
2.86
2.98
3.10
SIGMA R - DIAMOND SIGMA T - ASTERISK
SIGMA Z - CROSS TEMPERATURE - SQUARE
STRESS SCALE = 1.674X 10^4 PSI/INCH
TEMPERATURE SCALE = 6.7 DEGREES/INCH
RADIAL POSITION SCALE = 0.01 INCHES/INCH

FIGURE 24(d)

STRESS AND TEMPERATURE
VERSUS RADIAL POSITION
TIME = 0.750 CONF.4-120



SIGMA R - DIAMOND SIGMA T - ASTERISK
SIGMA Z - CROSS TEMPERATURE - SQUARE

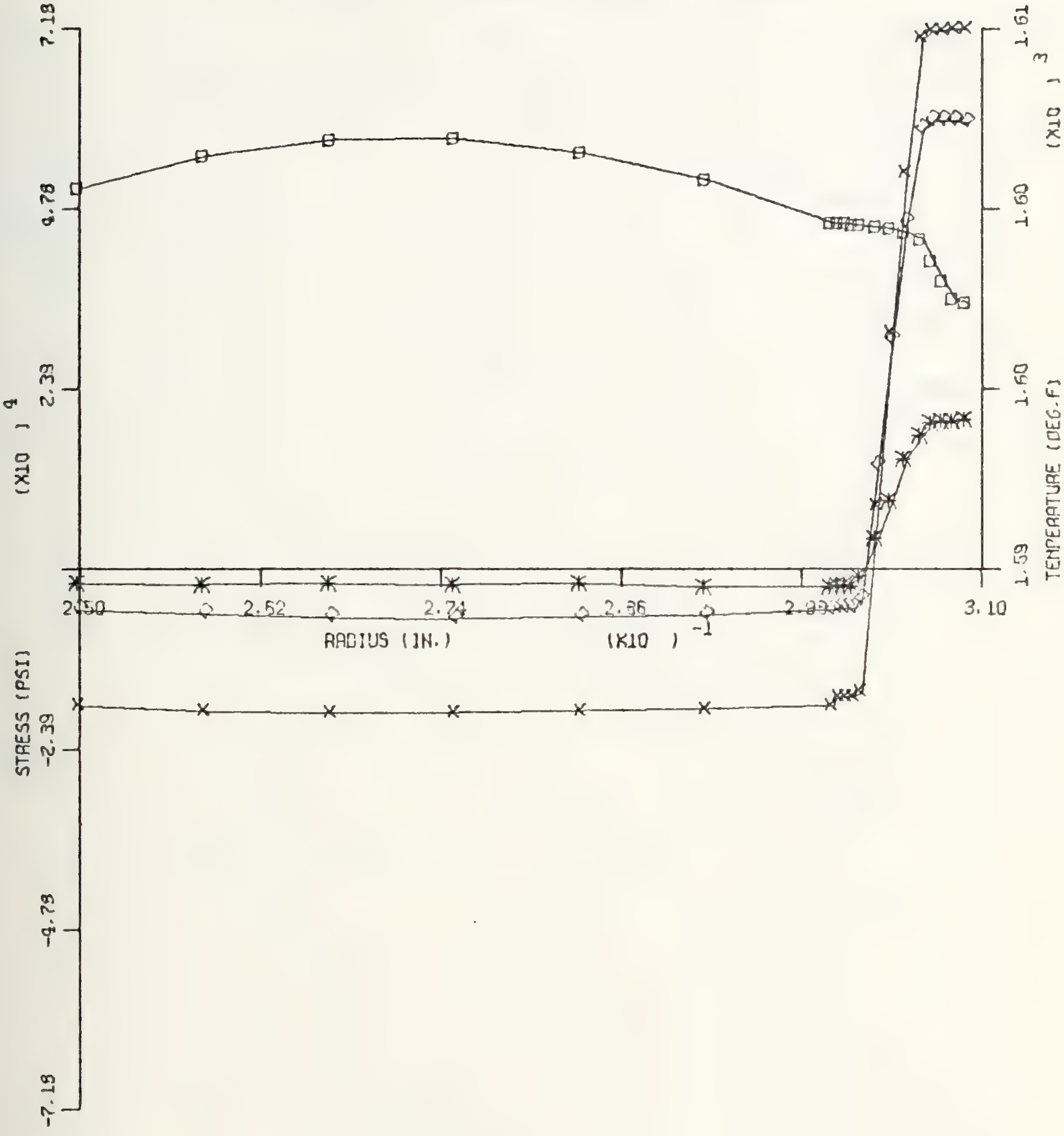
STRESS SCALE = 3.502×10^{4} PSI/INCH

TEMPERATURE SCALE = 3.3 DEGREES/INCH

RADIAL POSITION SCALE = 0.01 INCHES/INCH

FIGURE 24(e)

STRESS AND TEMPERATURE
VERSUS RADIAL POSITION
TIME = 0.800 CONF.4-120



SIGMA R - DIAMOND SIGMA T - ASTERISK
SIGMA Z - CROSS TEMPERATURE - SQUARE

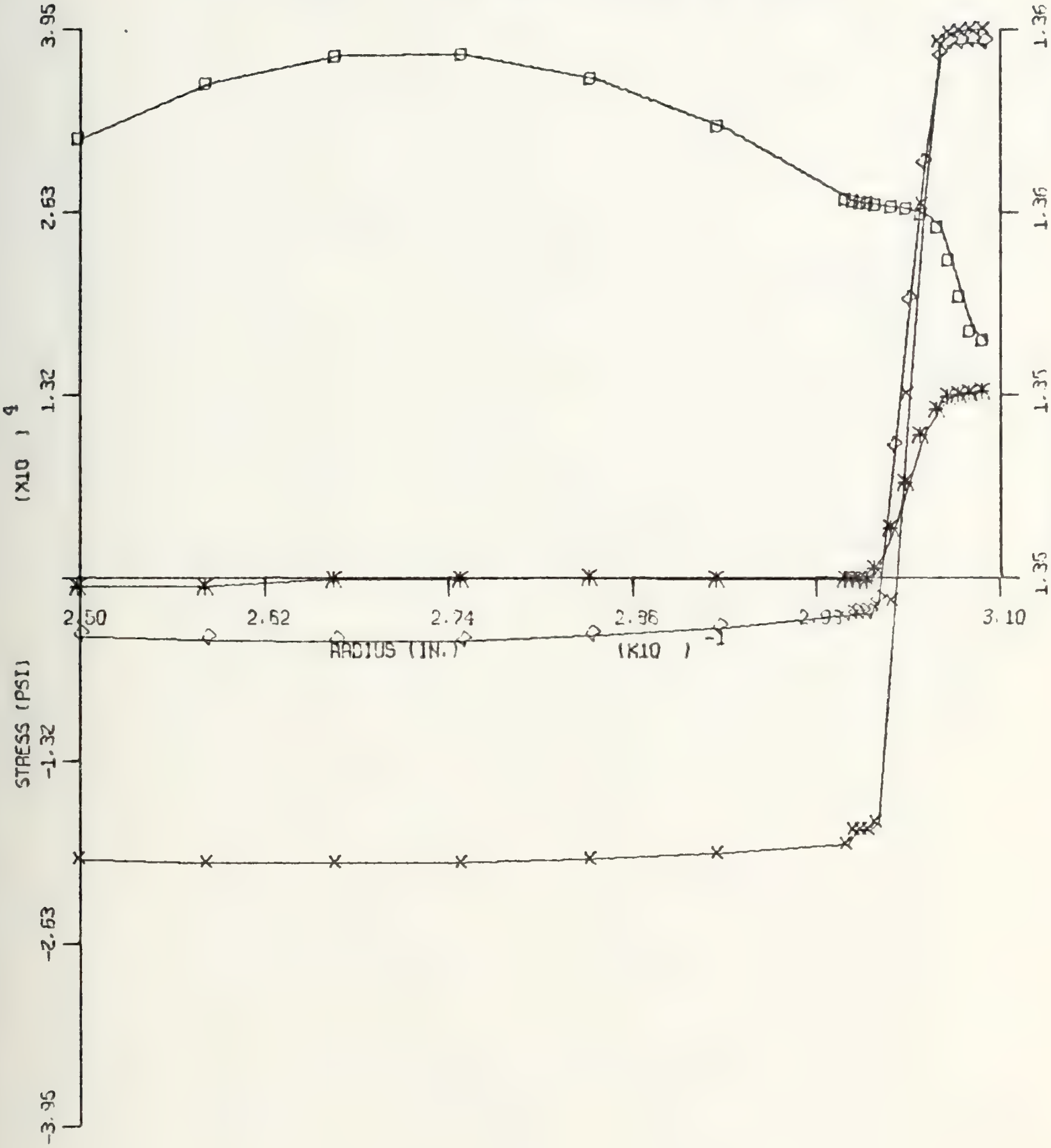
STRESS SCALE = 2.392K 10^4 PSI/INCH

TEMPERATURE SCALE = 6.7 DEGREES/INCH

RADIAL POSITION SCALE = 0.01 INCHES/INCH

FIGURE 24(f)

STRESS AND TEMPERATURE
VERSUS RADIAL POSITION
TIME = 0.850 CONF.4-120



SIGMA R - DIAMOND SIGMA T - ASTERISK
SIGMA Z - CROSS TEMPERATURE - SQUARE

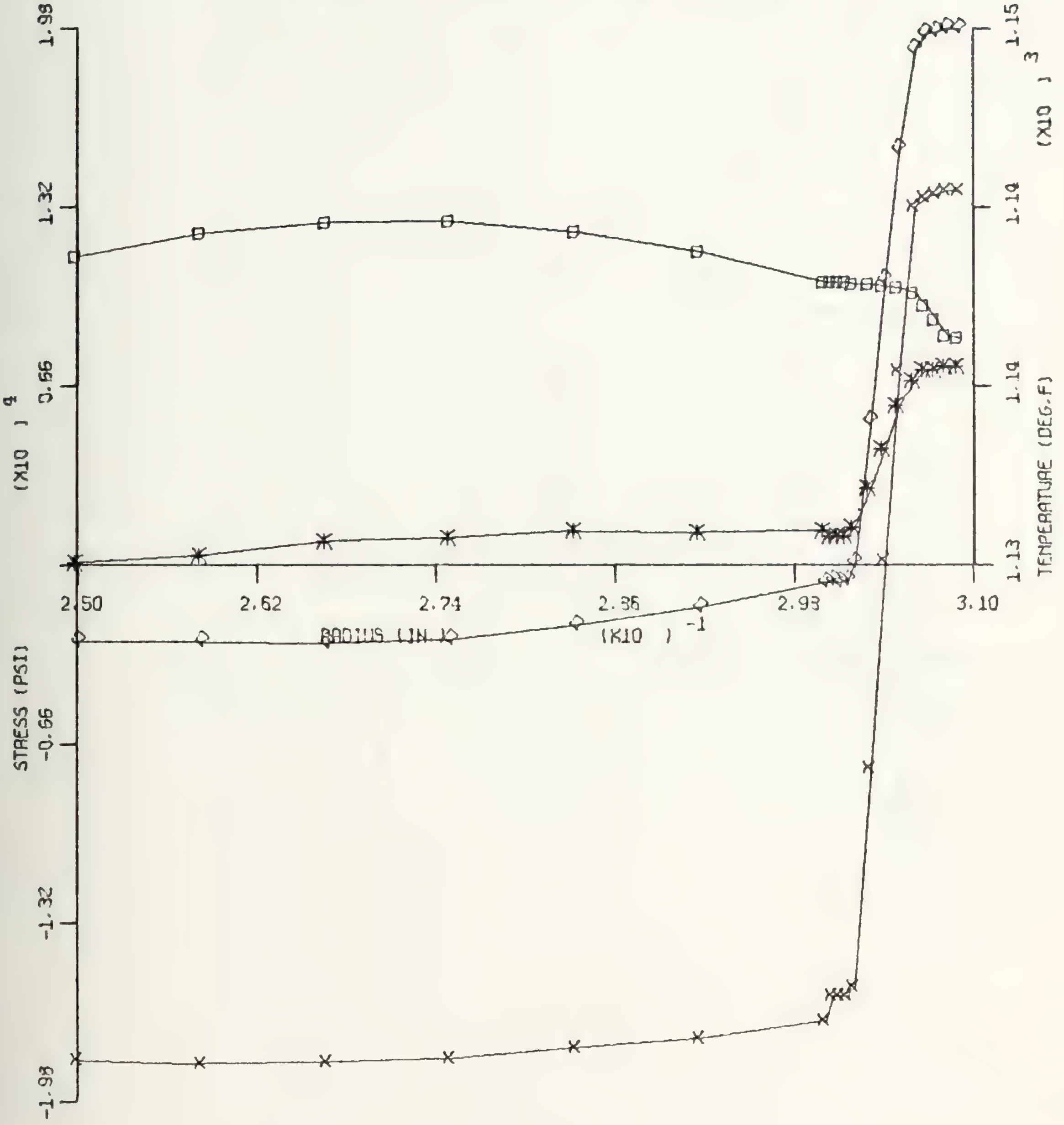
STRESS SCALE = 1.317X 10^{4} PSI/INCH

TEMPERATURE SCALE = 3.3 DEGREES/INCH

RADIAL POSITION SCALE = 0.01 INCHES/INCH

FIGURE 24(g)

STRESS AND TEMPERATURE
VERSUS RADIAL POSITION
TIME = 0.900 CONF.4-120



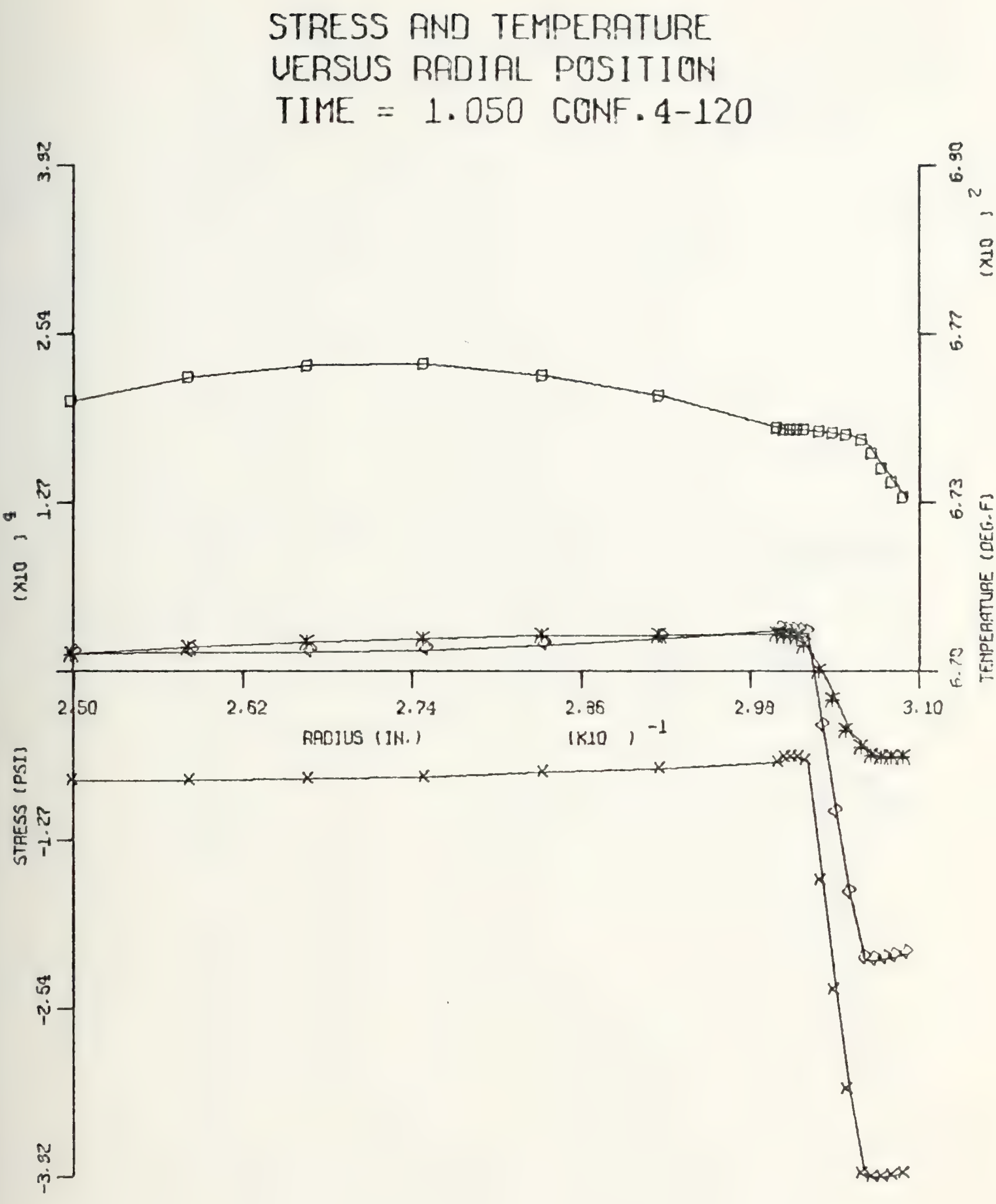
SIGMA R - DIAMOND SIGMA T - ASTERISK
SIGMA Z - CROSS TEMPERATURE - SQUARE

STRESS SCALE = 6.590X 10^3 PSI/INCH

TEMPERATURE SCALE = 6.7 DEGREES/INCH

RADIAL POSITION SCALE = 0.01 INCHES/INCH

FIGURE 24(h)



SIGMA R - DIAMOND SIGMA T - ASTERISK
SIGMA Z - CROSS TEMPERATURE - SQUARE

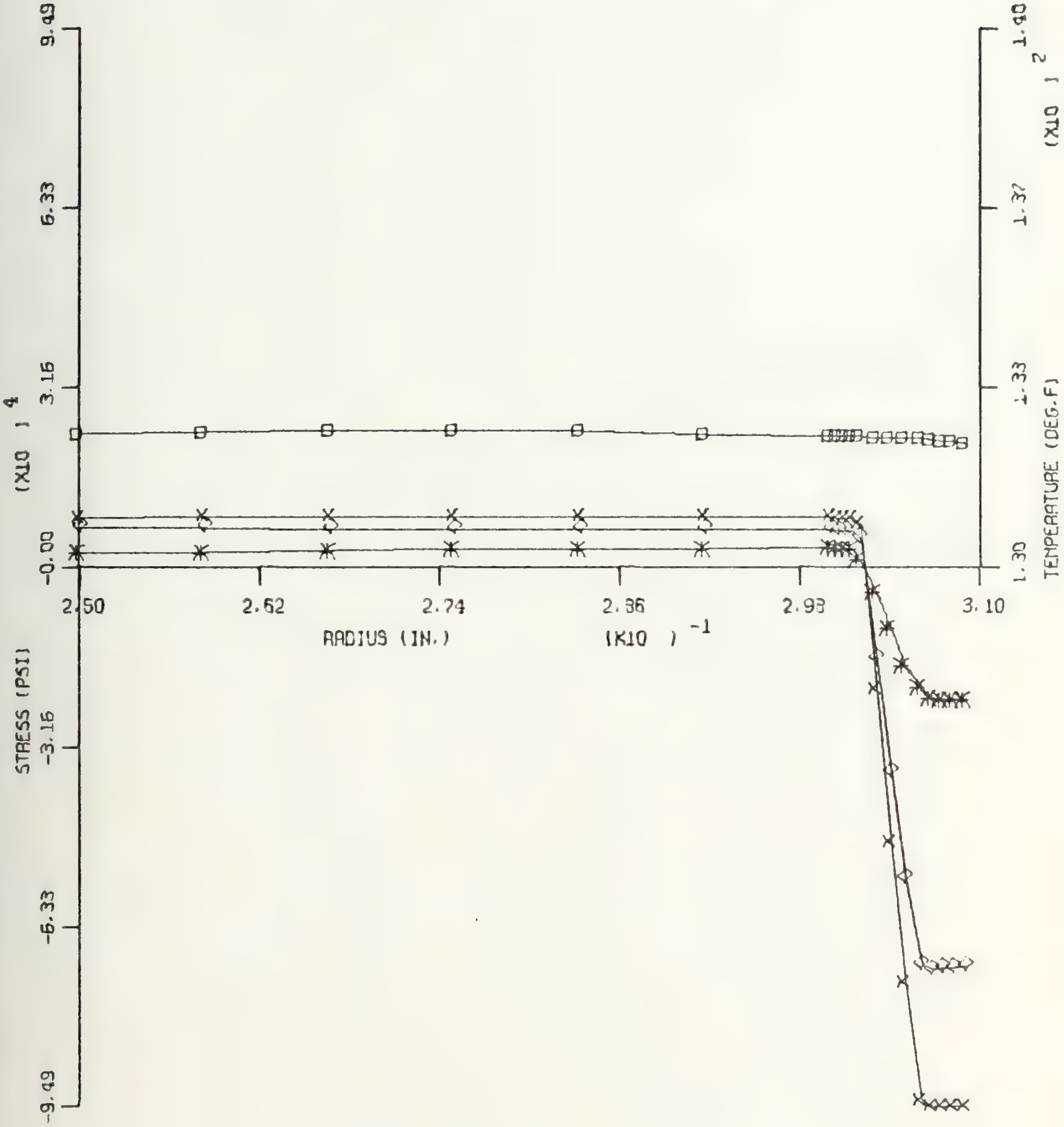
STRESS SCALE = 1.272K 10^4 PSI/INCH

TEMPERATURE SCALE = 3.3 DEGREES/INCH

RADIAL POSITION SCALE = 0.01 INCHES/INCH

FIGURE 24(i)

STRESS AND TEMPERATURE
VERSUS RADIAL POSITION
TIME = 1.650 CONF.4-120



SIGMA R - DIAMOND SIGMA T - ASTERISK
SIGMA Z - CROSS TEMPERATURE - SQUARE

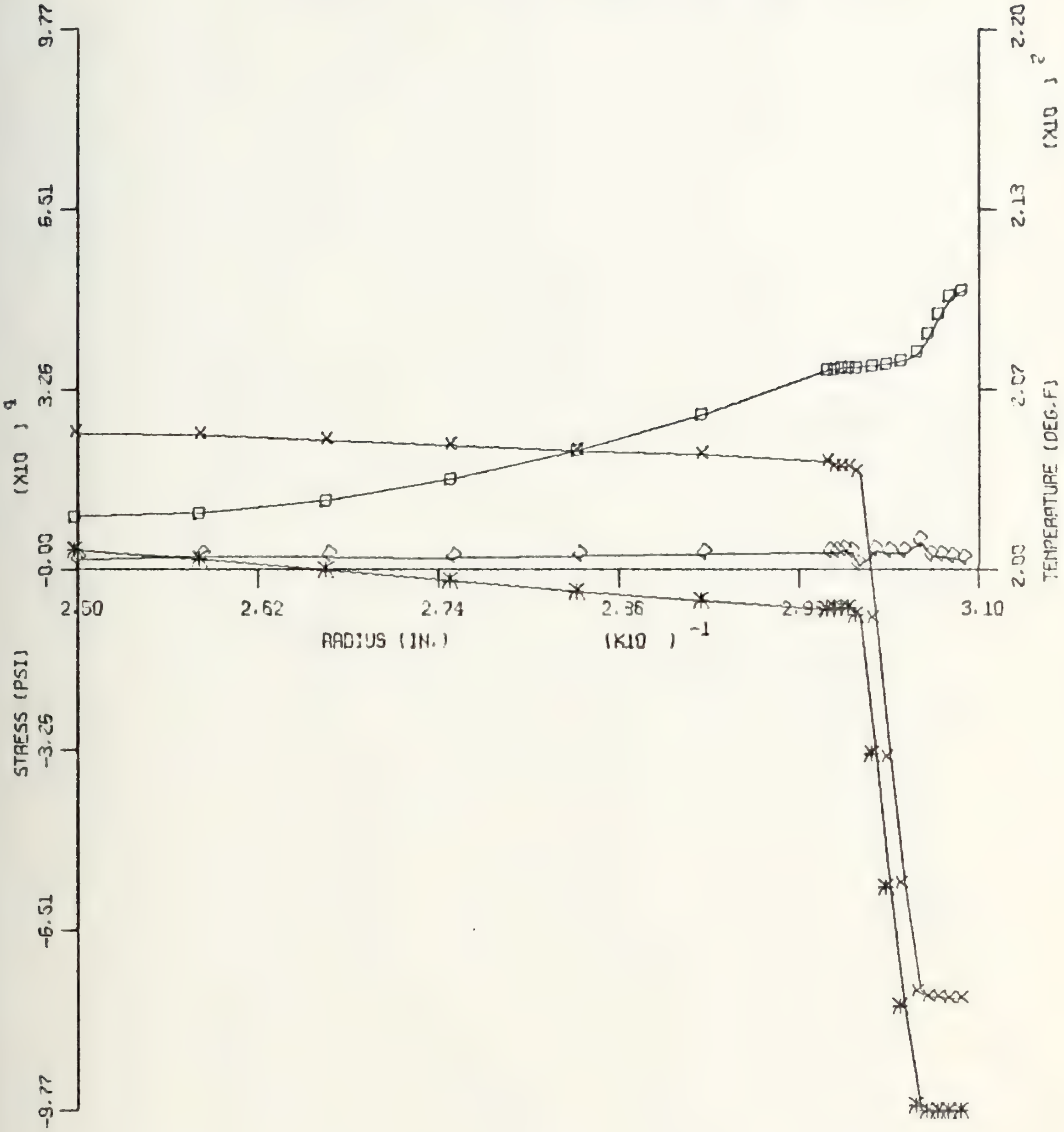
STRESS SCALE = 3.164X 10^4 PSI/INCH

TEMPERATURE SCALE = 3.3 DEGREES/INCH

RADIAL POSITION SCALE = 0.01 INCHES/INCH

FIGURE 24(j)

STRESS AND TEMPERATURE
VERSUS RADIAL POSITION
TIME = 0.050 CONF.4-180



SIGMA R - DIAMOND SIGMA T - ASTERISK
SIGMA Z - CROSS TEMPERATURE - SQUARE

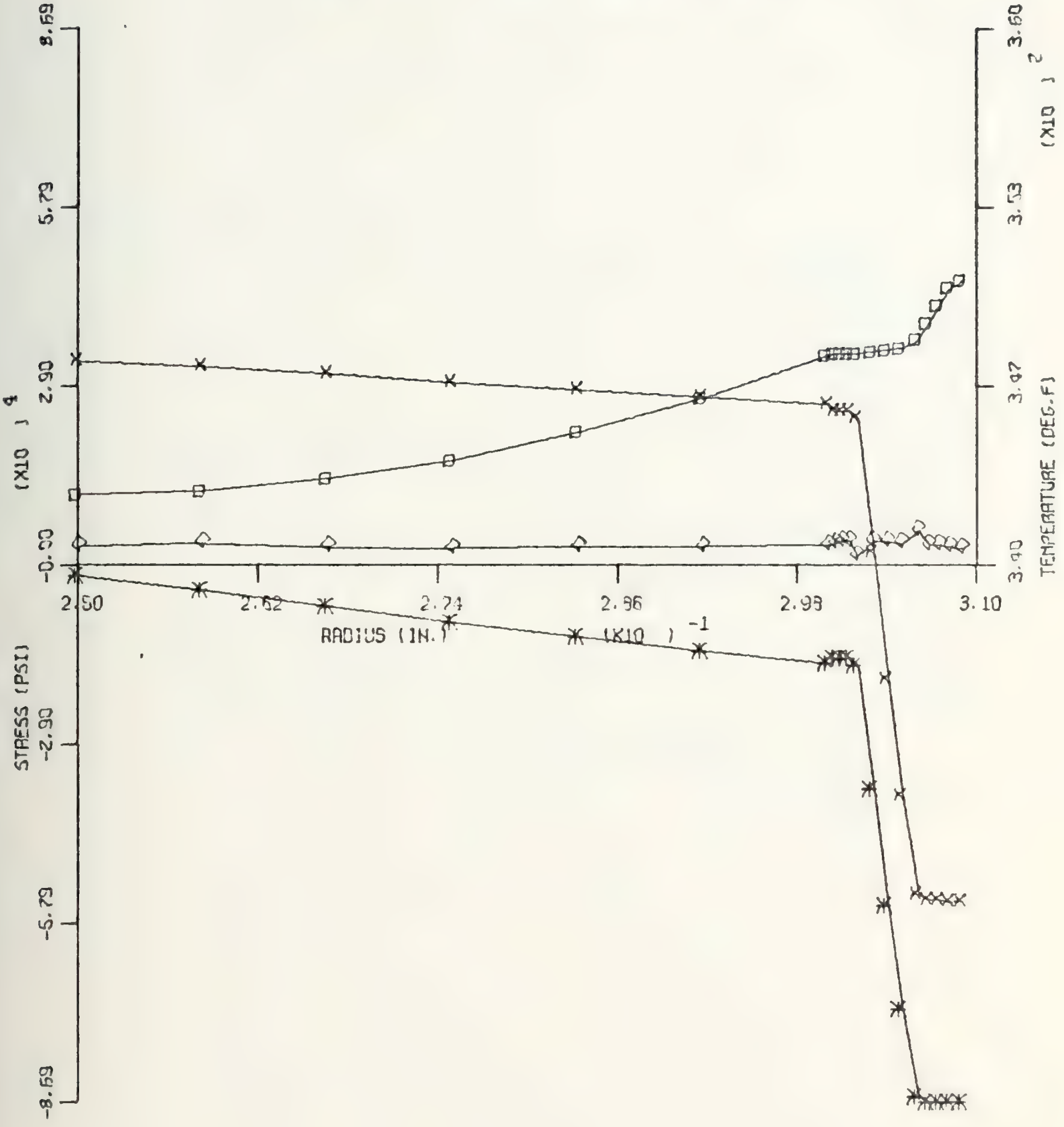
STRESS SCALE = 3.256X 10^4 PSI/INCH

TEMPERATURE SCALE = 6.7 DEGREES/INCH

RADIAL POSITION SCALE = 0.01 INCHES/INCH

FIGURE 25(a)

STRESS AND TEMPERATURE
VERSUS RADIAL POSITION
TIME = 0.100 CONF.4-180



SIGMA R - DIAMOND SIGMA T - ASTERISK
SIGMA Z - CROSS TEMPERATURE - SQUARE

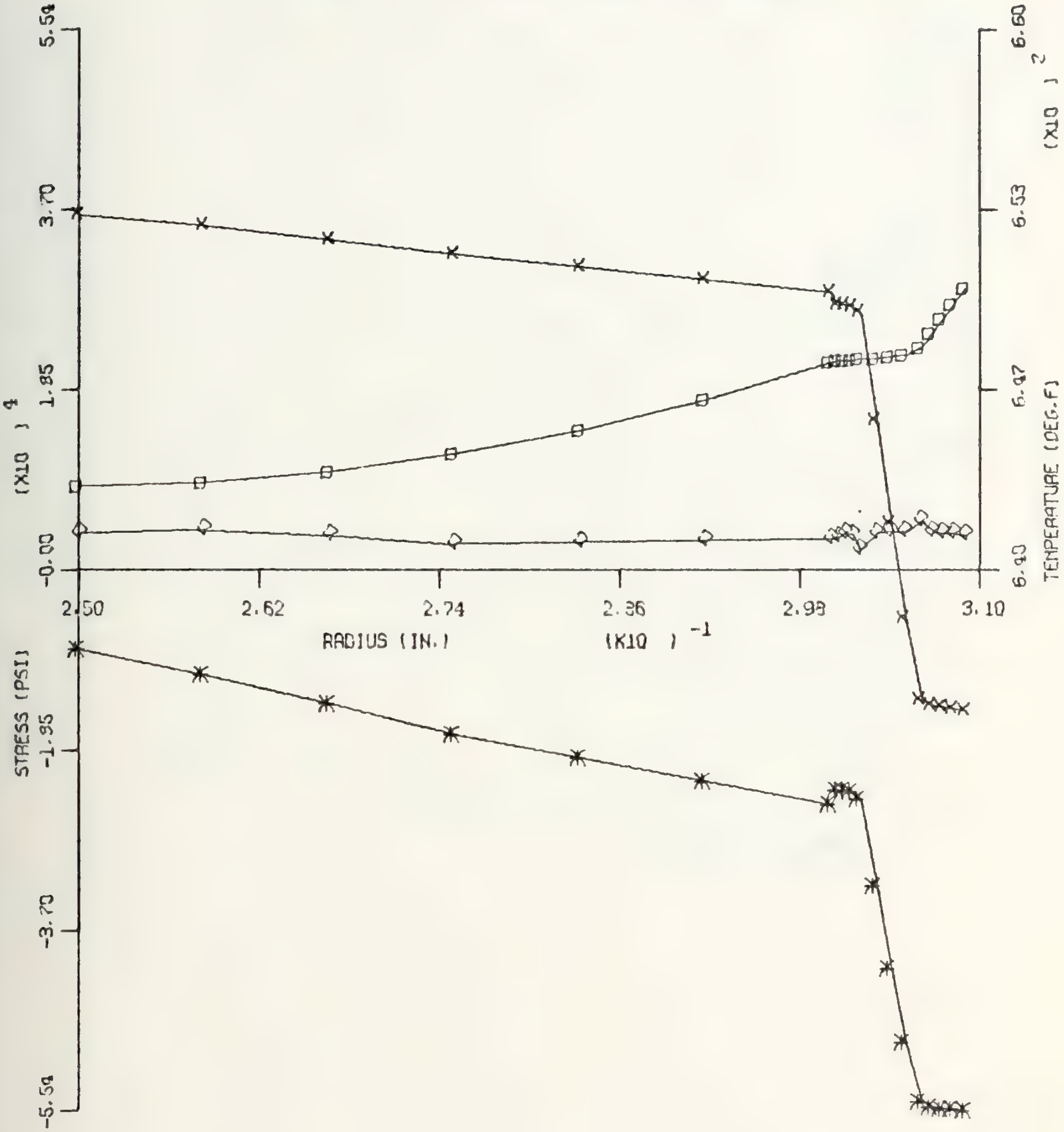
STRESS SCALE = 2.896K 10^4 PSI/INCH

TEMPERATURE SCALE = 6.7 DEGREES/INCH

RADIAL POSITION SCALE = 0.01 INCHES/INCH

FIGURE 25(b)

STRESS AND TEMPERATURE
VERSUS RADIAL POSITION
TIME = 0.200 CONF.4-180



SIGMA R - DIAMOND SIGMA T - ASTERISK
SIGMA Z - CROSS TEMPERATURE - SQUARE

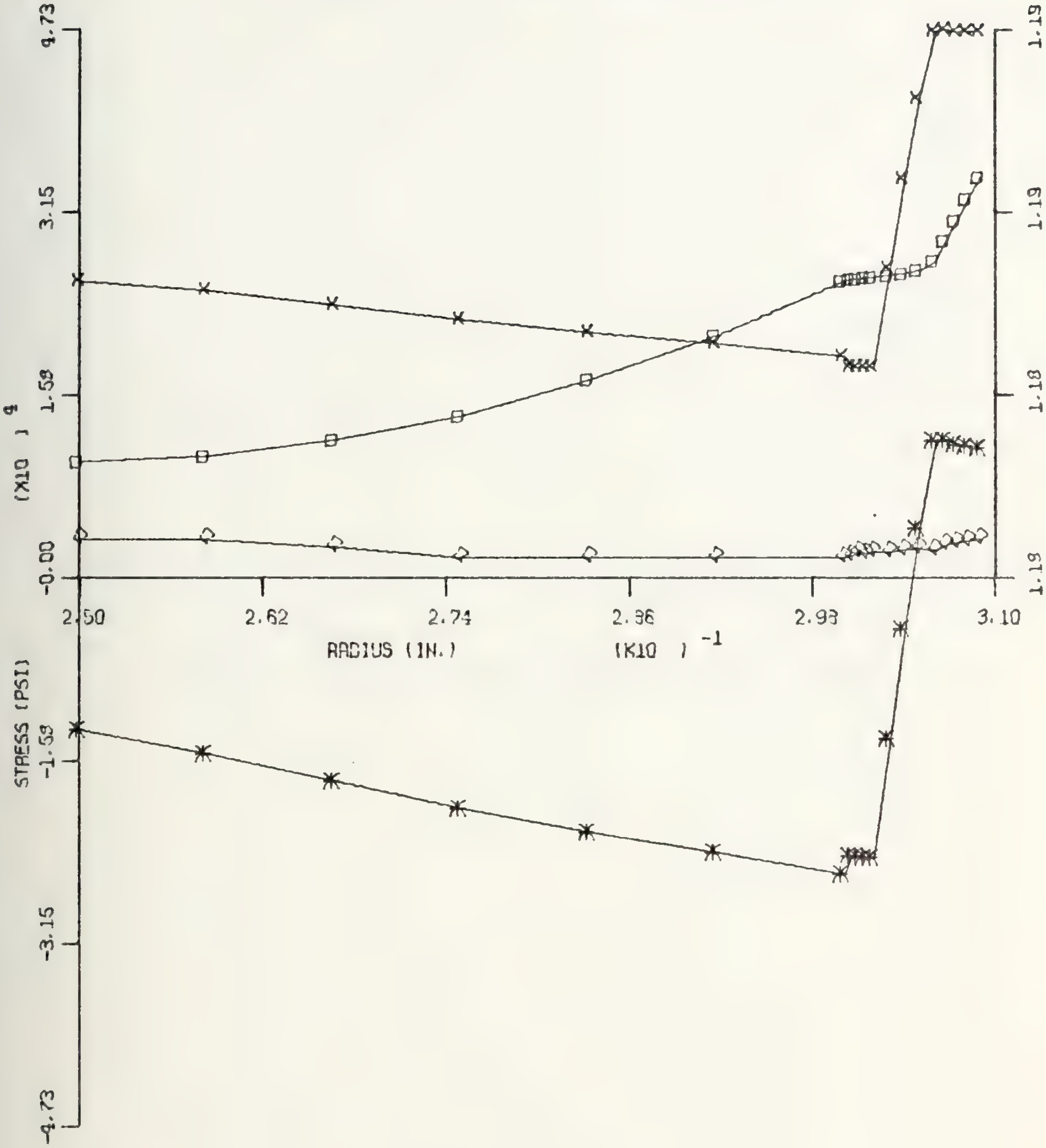
STRESS SCALE = 1.848X 10^4 PSI/INCH

TEMPERATURE SCALE = 6.7 DEGREES/INCH

RADIAL POSITION SCALE = 0.01 INCHES/INCH

FIGURE 25(c)

STRESS AND TEMPERATURE
VERSUS RADIAL POSITION
TIME = 0.400 CONF.4-180



SIGMA R - DIAMOND SIGMA T - ASTERISK
SIGMA Z - CROSS TEMPERATURE - SQUARE

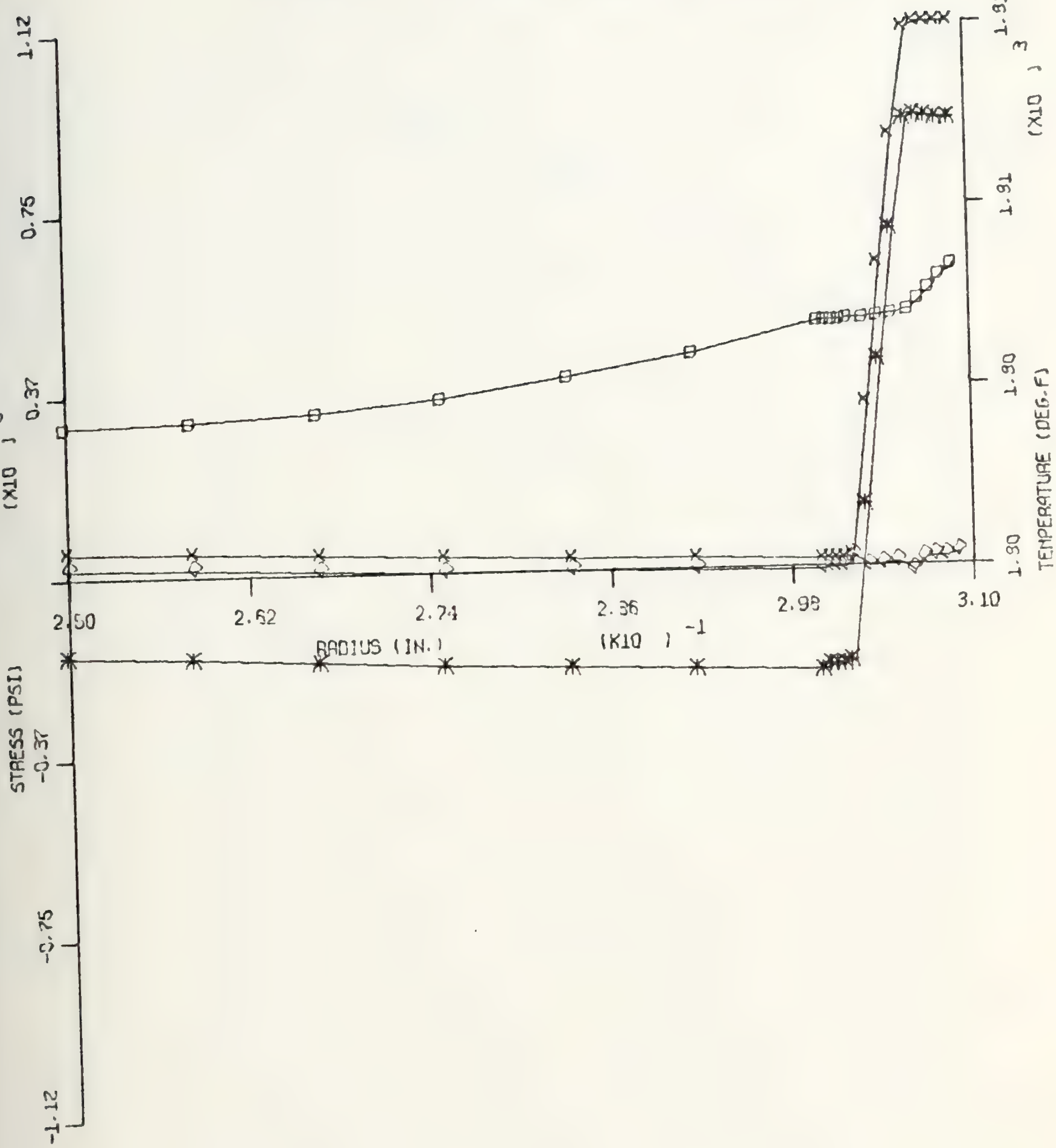
STRESS SCALE = 1.577X 10^4 PSI/INCH

TEMPERATURE SCALE = 3.3 DEGREES/INCH

RADIAL POSITION SCALE = 0.01 INCHES/INCH

FIGURE 25(d)

STRESS AND TEMPERATURE
VERSUS RADIAL POSITION
TIME = 0.750 CONF.4-180



SIGMA R - DIAMOND SIGMA T - ASTERISK
SIGMA Z - CROSS TEMPERATURE - SQUARE

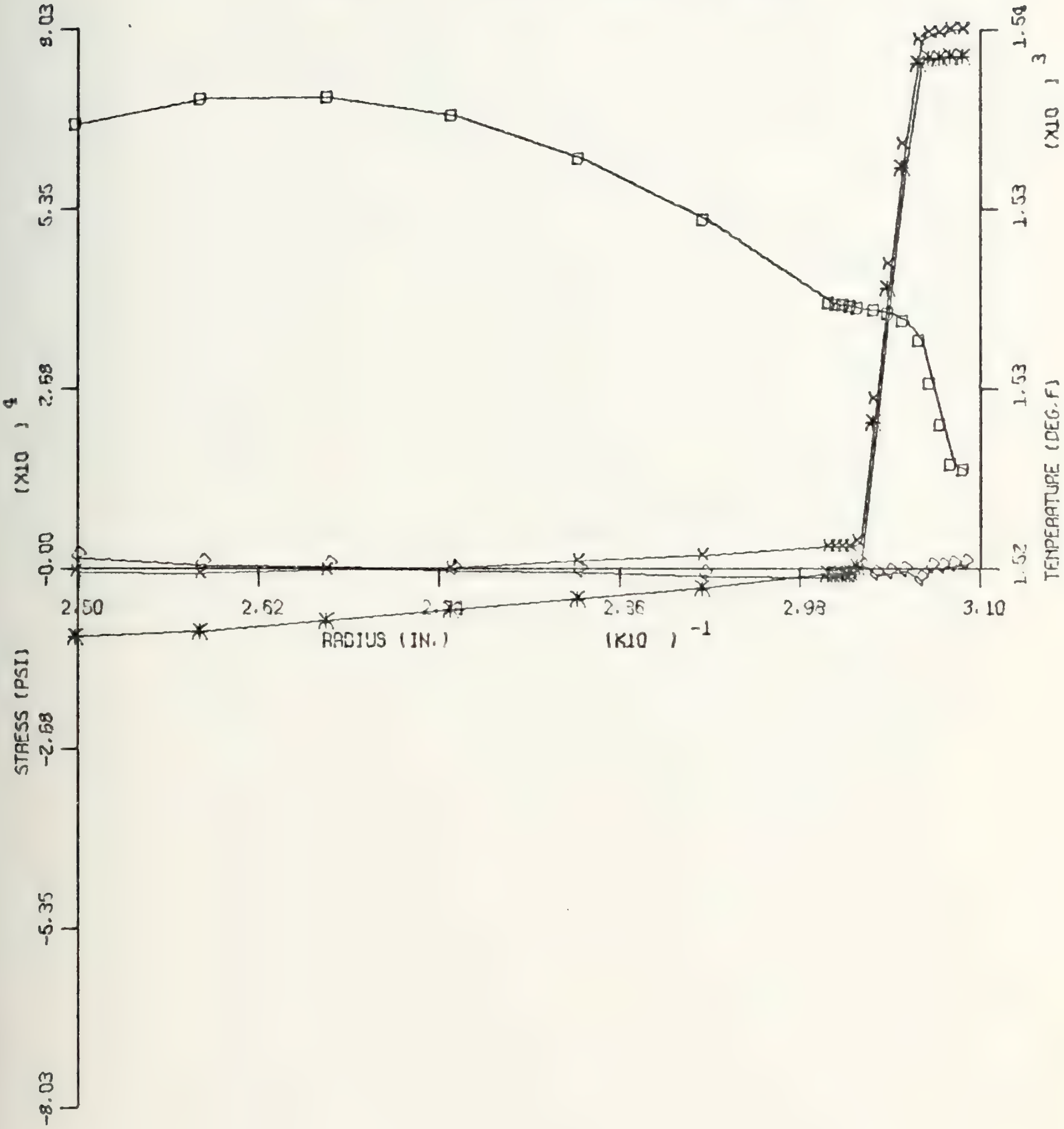
STRESS SCALE = 3.739K 10 4 PSI/INCH

TEMPERATURE SCALE = 3.3 DEGREES/INCH

RADIAL POSITION SCALE = 0.01 INCHES/INCH

FIGURE 25(e)

STRESS AND TEMPERATURE
VERSUS RADIAL POSITION
TIME = 0.800 CONF.4-180



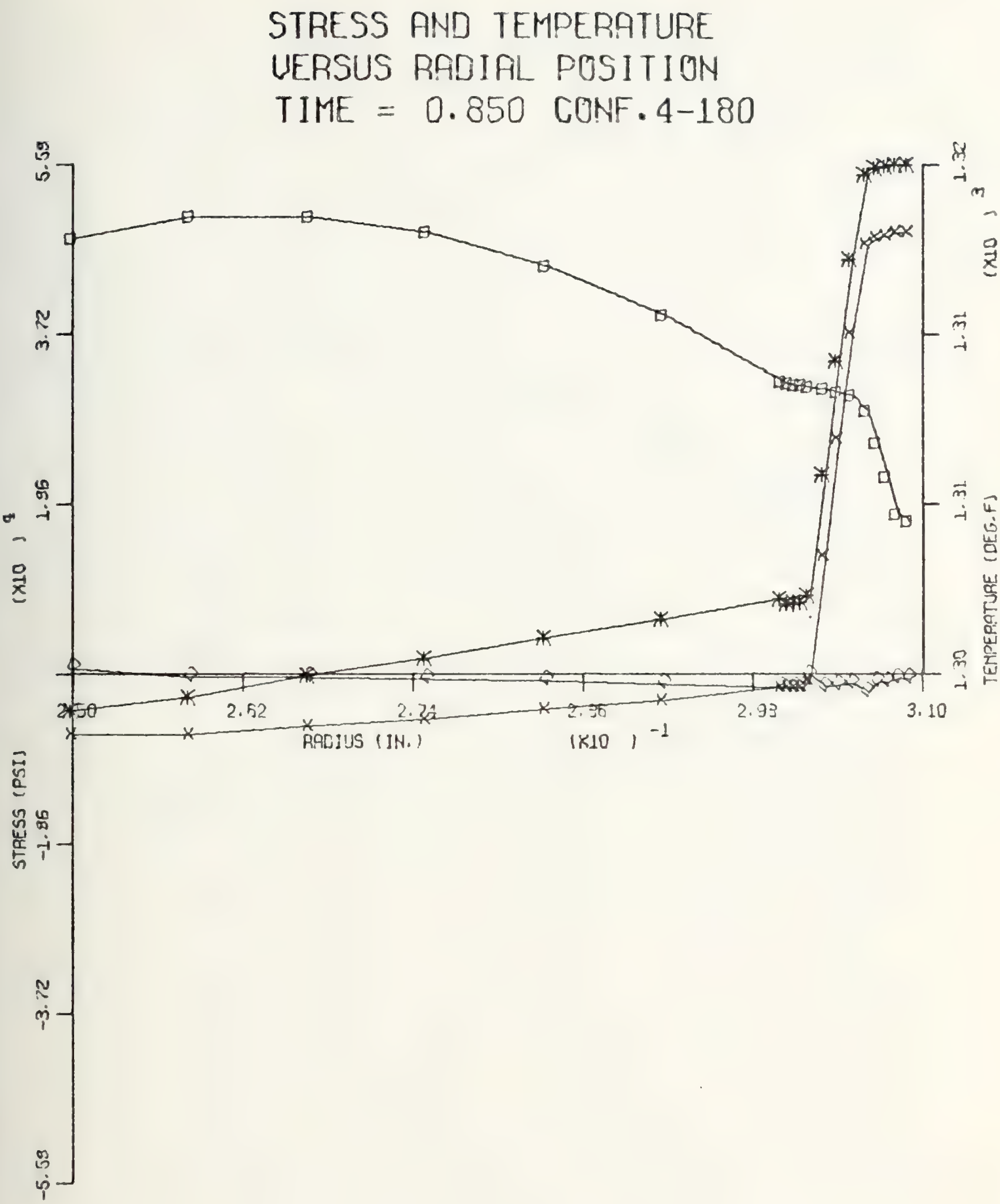
SIGMA R - DIAMOND SIGMA T - ASTERISK
SIGMA Z - CROSS TEMPERATURE - SQUARE

STRESS SCALE = 2.675 X 10^4 PSI/INCH

TEMPERATURE SCALE = 6.7 DEGREES/INCH

RADIAL POSITION SCALE = 0.01 INCHES/INCH

FIGURE 25(f)



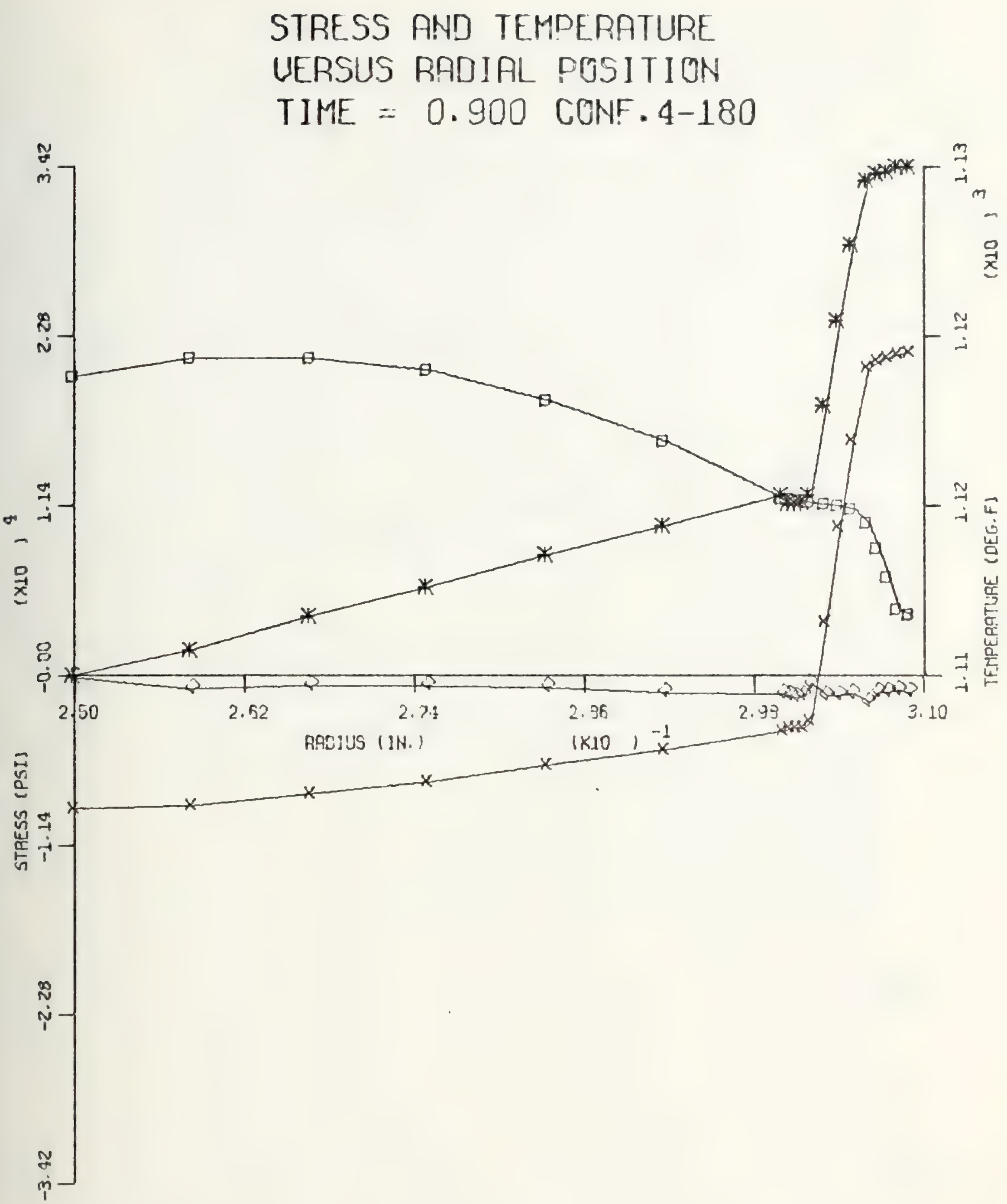
SIGMA R - DIAMOND SIGMA T - ASTERISK
SIGMA Z - CROSS TEMPERATURE - SQUARE

STRESS SCALE = 1.858X 10^4 PSI/INCH

TEMPERATURE SCALE = 6.7 DEGREES/INCH

RADIAL POSITION SCALE = 0.01 INCHES/INCH

FIGURE 25(g)



SIGMA R - DIAMOND SIGMA T - ASTERISK
SIGMA Z - CROSS TEMPERATURE - SQUARE

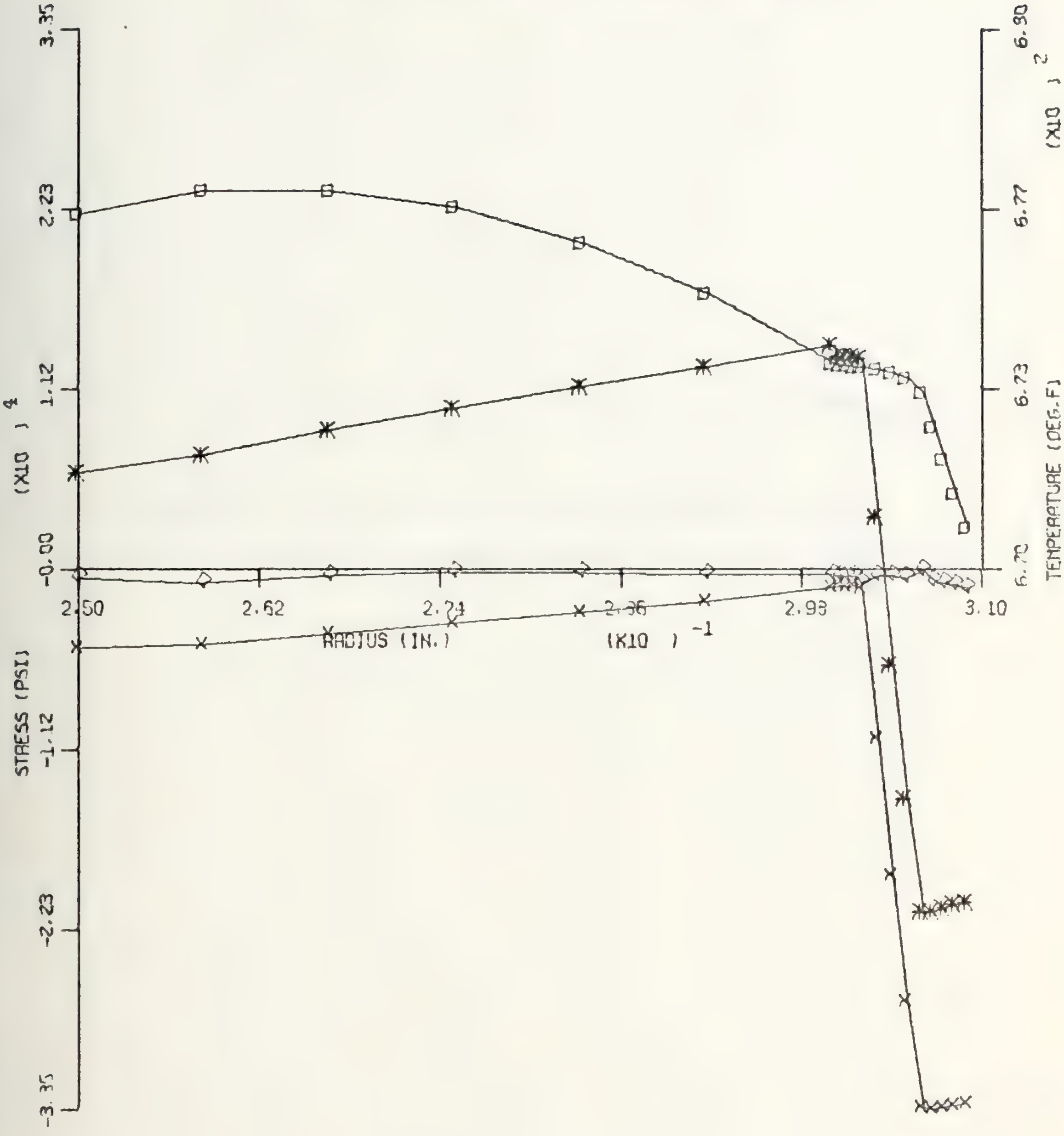
STRESS SCALE = 1.141 X 10^4 PSI/INCH

TEMPERATURE SCALE = 6.7 DEGREES/INCH

RADIAL POSITION SCALE = 0.01 INCHES/INCH

FIGURE 25(h)

STRESS AND TEMPERATURE
VERSUS RADIAL POSITION
TIME = 1.050 CONF.4-180



SIGMA R - DIAMOND SIGMA T - ASTERISK
SIGMA Z - CROSS TEMPERATURE - SQUARE

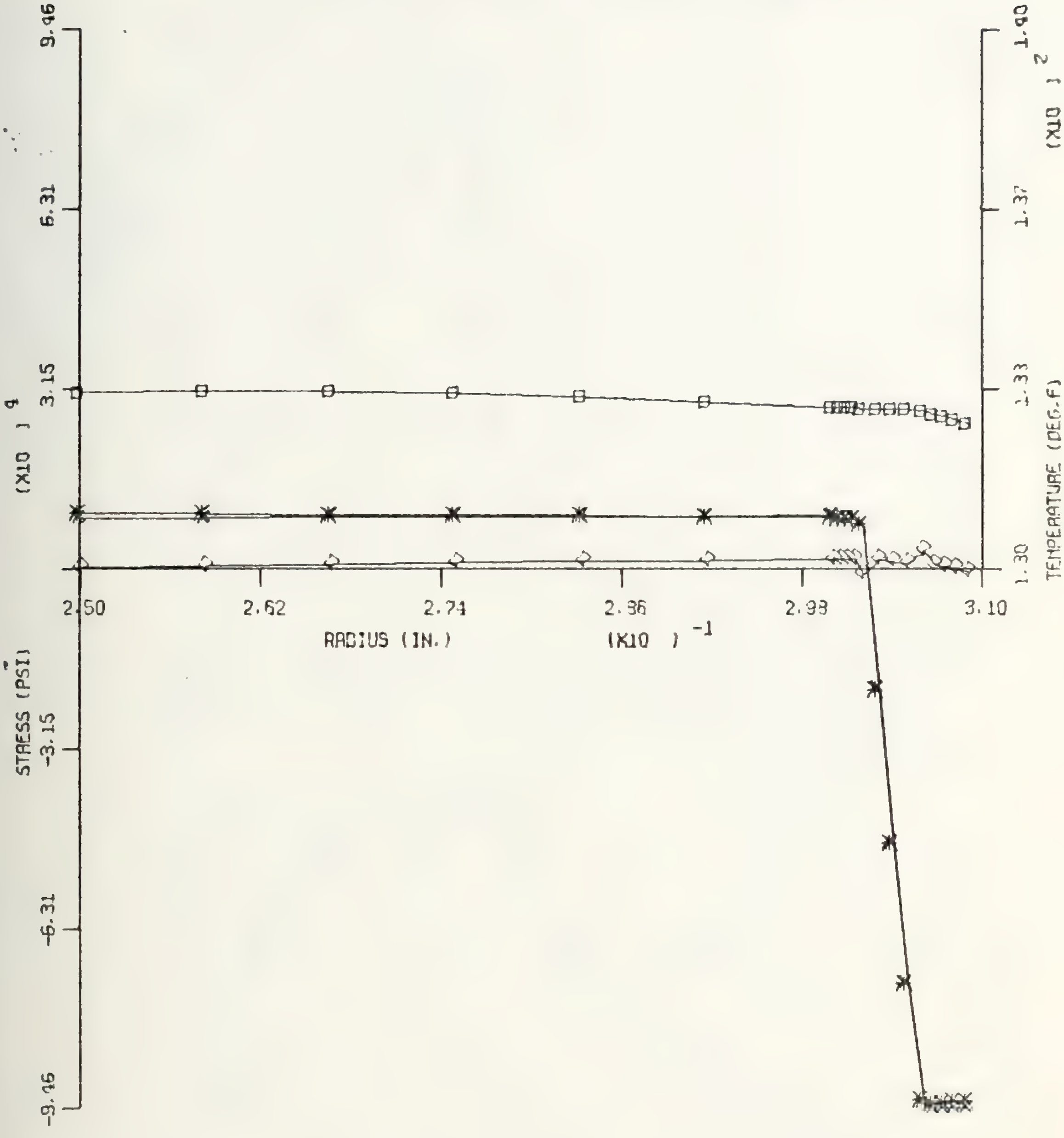
STRESS SCALE = 1.117X 10^4 PSI/INCH

TEMPERATURE SCALE = 3.3 DEGREES/INCH

RADIAL POSITION SCALE = 0.01 INCHES/INCH

FIGURE 25(i)

STRESS AND TEMPERATURE
VERSUS RADIAL POSITION
TIME = 1.650 CONF.4-180



SIGMA R - DIAMOND SIGMA T - ASTERISK
SIGMA Z - CROSS TEMPERATURE - SQUARE

STRESS SCALE = 3.153X 10^4 PSI/INCH

TEMPERATURE SCALE = 3.3 DEGREES/INCH

RADIAL POSITION SCALE = 0.01 INCHES/INCH

FIGURE 25(j)

MAXIMUM AND MINIMUM PRINCIPAL STRESSES
VERSUS TIME
CONFIGURATION NUMBER FOUR

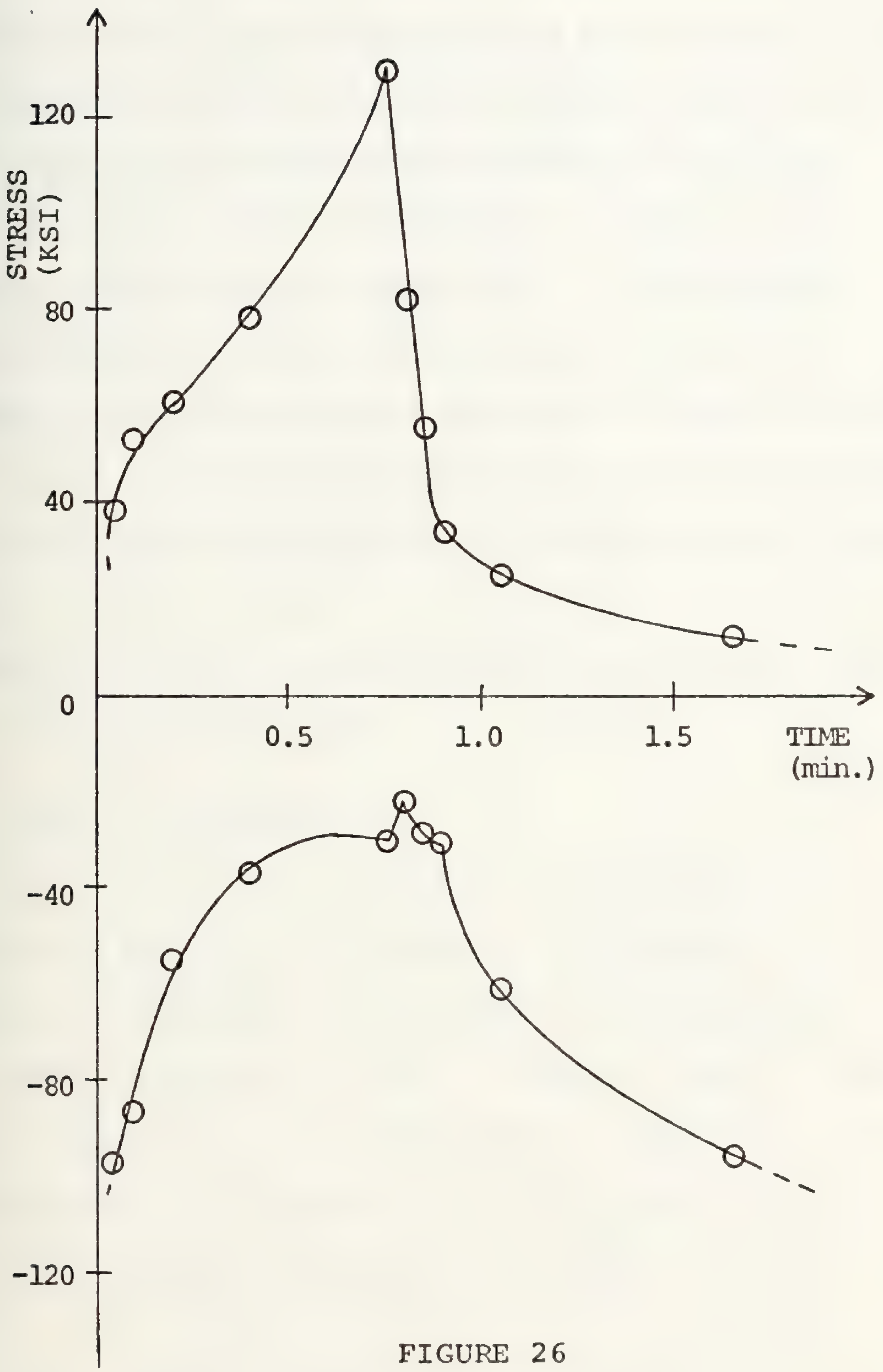


FIGURE 26

VI. CONCLUSIONS AND RECOMMENDATIONS

The thermal stress analysis of the various cylinder configurations showed that the maximum stress levels of the four coating arrangements were approximately equal. The maximum stress level was excessive enough to cause failure of the coatings and was dependent on the deposition temperature and the maximum temperature distribution. The abrupt change in the stress levels between the substratum and the outer coatings of configurations one and two was reduced by the use of the graded layers in configurations three and four.

The following recommendations are suggested for future development and analysis.

A. MATERIAL PROPERTIES

An extensive investigation of the temperature dependent properties should be completed. In addition to the various individual materials, the graded layer properties must be thoroughly understood before a complete design analysis can be completed. All creep data for these materials must be obtained. If possible, material properties for the compressive stress state should be obtained.

B. VARIATIONS TO THE PROBLEM

The following situations should be investigated.

1. Other geometries, such as wedges and turbine blades, would exhibit different temperature and stress distributions. Investigation of these geometries may show lower stress levels which would be less susceptible to failure. These investigations should be made.

2. The temperature distribution and maximum stress levels are directly related to the thickness of the zirconia coating. This coating depth should be varied to determine an optimal thickness for the minimization of the stress levels.

3. Since the deposition temperature affects the maximum tensile stresses, more investigation to determine the most appropriate temperature for coating application should be made. More information about the compressive state will also allow a better evaluation of the effect of the deposition temperature selection.

4. Although changes in the material property variation of the graded layer produced minor changes in the stress levels, the effect of the thickness of the graded layer should be investigated.

5. Since thermal load conditions are only part of the total load conditions on a turbine blade, an investigation of the effect of body forces on the stress levels in the blade should be completed. The program used in this analysis allows the specification of these transient body forces.

PART II. TRANSIENT THERMAL STRESS ANALYSIS

The classical equations governing thermal stress analysis are coupled to the heat conduction equations, however, in most studies of this topic including this one, the stress analysis and the heat conduction are separated. The analysis of the transient thermal stress condition is performed in two steps. In the first one, the unsteady heat equation is solved for the temperature distribution. Then, using this temperature distribution, the equations of elasticity can be solved to determine the stress distribution. Analytically this solution process is possible only in the simplest of problems. However, through the use of the finite element techniques, an approximate solution can be obtained for more realistic problems.

The computer program which was used as a basis to solve the heat conduction equation was written by Lew [7]. Several modifications were made to this program as described in Section I. The computer program used as a basis to solve the stress problem was written by Leonidas [8]. Several changes, as described in Section II, were also made to this program.

Since three dimensional finite element analysis generates such a voluminous amount of information, an additional post-processing program was written to condense the relevant information.

All programs were written using Fortran IV G Level language and solutions were obtained using the IBM 360/67 computer system.

I. TEMPERATURE DISTRIBUTION

The unsteady heat conduction equation, expressed in rectangular coordinates, is

$$\frac{\partial}{\partial x}(k_x \frac{\partial T}{\partial x}) + \frac{\partial}{\partial y}(k_y \frac{\partial T}{\partial y}) + \frac{\partial}{\partial z}(k_z \frac{\partial T}{\partial z}) + Q = \rho C \frac{\partial T}{\partial t} \qquad (2)$$

This equation is subject to the following boundary conditions:

$$T = T_B \quad \text{on the boundary or}$$

$$k_x \frac{\partial T}{\partial x} l_x + k_y \frac{\partial T}{\partial y} l_y + k_z \frac{\partial T}{\partial z} l_z + q_s + h(T-T_f) = 0 \qquad (3)$$

on the boundary

and a specified initial temperature distribution. The symbols used in these equations have the following definitions:

Q	- rate of internal energy generation per unit volume
ρ	- density
C	- specific heat
k_x, k_y, k_z	- thermal conductivity in the x,y, and z directions
l_x, l_y, l_z	- the direction cosines at a point on the boundary
q_s	- external heat flux
h	- surface heat transfer coefficient
T_f	- fluid temperature

As described in Ref. 9, the unsteady heat equation can be transformed into a discretized form by the finite element technique. Using the symbol [ ] for an n x n matrix, < > for a row vector, and { } for a column vector,

$$[H]\ \{T\} + [C]\ \{\dot{T}\} + \{F\} = 0 \tag{4}$$

where the temperature T of equation (2) has been expressed as the product of the nodal shape functions and the nodal temperature values.

$$T = \langle N\rangle\ \{T\} \tag{5}$$

The time derivative of the temperature $\frac{\partial T}{\partial t}$ has been expressed as

$$\frac{\partial T}{\partial t} = \dot{T} = \langle N\rangle\ \{\dot{T}\} \tag{6}$$

The matrices [H], [C], {F} are defined at the element level by

$$[H]_e = \int_{V_e} (k_x\{\frac{\partial N}{\partial x}\}\langle\frac{\partial N}{\partial x}\rangle + k_y\{\frac{\partial N}{\partial y}\}\langle\frac{\partial N}{\partial y}\rangle + k_z\{\frac{\partial N}{\partial z}\}\langle\frac{\partial N}{\partial z}\rangle)\,dx\ dy\ dz \tag{7}$$

$$[C]_e = \int_{V_e} \{N\}\ \rho C\ \langle N\rangle\ dx\ dy\ dz \tag{8}$$

$$\{F\}_e = -\int_{V_e} Q\{N\}dx\, dy\, dz + \int q\{N\}dS_e + \left(\int_{S_e} \{N\}h<N>dS_e\right)\{T_i\} \tag{9}$$

where

$$q = q_s - h\, T_f$$

The program developed by Lew [7] solves the discretized equation subject to the listed boundary conditions. In order to adapt the program to the thermal stress problem, several alterations were incorporated.

A. TIME INTEGRATION SCHEME

Reference 7 provides numerous time integration schemes to deal with the first time derivative of the temperature. In this study only the best integration schemes were retained. A five point backward difference formula was used to approximate the time derivative. For the ith time step, this approximation can be expressed as

$$\left(\frac{\partial T}{\partial t}\right)_i = \dot{T}_i = \frac{1}{12\Delta t}(3T_{i-4} - 16T_{i-3} + 36T_{i-2} - 48T_{i-1} + 25T_i) \tag{10}$$

where Δt is the specified time step. The truncation error for this expression is $+\frac{1}{5}(\Delta t)^4 \frac{\partial^5 T}{\partial t^5}$.

If equation (10) is written in vector form and substituted into equation (4), the final form of the discretized heat equation is obtained.

$$(12\Delta t[H] + 25[C])\cdot\{T_i\} =$$

$$48\{T_{i-1}\} - 36\{T_{i-2}\} + 16\{T_{i-3}\} - 3\{T_{i-4}\} - 12\Delta t\{F\} \tag{11}$$

If [H], [C], {F}, Δt and the four preceding values of $\{T_i\}$ are known, the system of simultaneous equations is solved for the next set of $\{T_i\}$.

The initial four values for the temperature distribution is obtained by using the trapezoid method for a specified number of steps at a small Δt. After each n steps at Δt, the values of $\{T_i\}$ are saved. When three sets of $\{T_i\}$ plus the initial known distribution are available, the transient solution using a time step of $(n\Delta t)$ can proceed. For example, an initial step size of 0.001 taken for 150 steps results in a step size of 0.05 for the remainder of the transient problem.

Using the solution technique described above, the transient temperature distribution for an infinite hollow cylinder were determined. The problem characteristics are:

Inside diameter	= 0.5 inches
Outside diameter	= 0.6 inches
Initial temperature	= 932 deg. F
Internal and external heat transfer coefficients	= 25.0 BTU/hr-sq.ft.-deg.F
Internal Fluid temperature	= 896 deg. F
External Fluid temperature	= 1994 deg. F
Initial time step size	= 0.001 min.
Number of initial steps	= 150

Results of the approximate solution and the exact solution are given in Table 3.

B. NODAL MATERIAL PROPERTIES

The usual method of specifying properties to be constant throughout a finite element becomes troublesome when composite materials have continuously varying properties. Many separate elements would be required in order to adequately represent a structure which has material properties that are continuous functions of spatial coordinates.

In order to alleviate the problem, several routines in the basic program of Ref. 7 were altered to accommodate nodal properties in addition to element properties. A property P is expressed as $P = N_i\ P_i$, where P_i is a vector of nodal values and N_i are the same shape functions used in equation (5) for the temperature variable. The matrices [H], [C], and {F} are formed by the same integration techniques as used in the basic program. However, the properties are no longer constant during the integration, and must be re-evaluated at each integration point.

As an example, consider steady state heat flow through a cylinder in which the thermal conductivity varies continuously along a radius. As proposed above, the development of a twenty nodal point element could represent exactly any linear or quadratic variation of material properties. Table 4 displays the results of such a problem as compared to the exact solution. The column heading "Element Property

TABLE 3

TRANSIENT TEMPERATURE OF THE INFINITE HOLLOW CYLINDER

Exact Solution (Deg. F)

Radius (in)	t = 0.25 (min)	0.5	1.0	3.0
0.25	1277.165	1409.796	1479.776	1491.715
0.26	1278.061	1410.916	1481.013	1492.977
0.27	1279.104	1412.062	1482.214	1494.182
0.28	1280.291	1413.236	1483.381	1495.348
0.29	1281.618	1414.438	1484.518	1496.473
0.30	1283.081	1415.669	1485.626	1497.561

Finite Element Solution

Using 5-Point Finite Difference Method (Deg. F)

Radius (in)	t = 0.25 (min)	0.5	1.0	3.0
0.25	1277.220	1409.897	1479.879	1491.810
0.26	1278.116	1411.016	1481.116	1493.068
0.27	1279.160	1412.163	1482.316	1494.278
0.28	1280.347	1413.336	1483.483	1495.443
0.29	1281.674	1414.538	1484.612	1496.568
0.30	1283.138	1415.769	1485.726	1497.654

TABLE 4

STEADY HEAT FLOW THROUGH A HOLLOW CYLINDER

Inside Diameter	0.5 in.
Outside Diameter	0.6 in.
Inside Surface Temperature	500.0 °F
Outside Surface Temperature	1000.0 °F

Thermal Conductivity Variation in Outward Radial Direction:

(a) For linear function, 0.05 to 0.25 BTU/in-min-°F

(b) For quadratic function, 0.2025 to 0.225625 (at mid-radius) to 0.25 BTU/in-min-°F

Linear Variation of Thermal Conductivity
(Two Radial Elements)

Radius (in.)	Exact Solution (deg. F)	Element Property Solution (deg. F)	Nodal Property Solution (deg. F)
0.25	500.0	500.0	500.0
0.2625	575.884	590.871	576.502
0.275	669.145	677.510	670.301
0.2875	794.697	842.256	796.982
0.3	1000.0	1000.0	1000.0

Quadratic Variation of Thermal Conductivity
(Two Radial Elements)

Radius (in.)	Exact Solution (deg. F)	Element Property Solution (deg. F)	Nodal Property Solution (deg. F)
0.25	500.0	500.0	500.0
0.2625	623.879	627.602	624.213
0.275	748.213	749.253	748.803
0.2875	873.441	877.342	873.747
0.3	1000.0	1000.0	1000.0

Solution" identifies the results obtained by specifying properties constant over an element. The column heading "Nodal Property Solution" identifies the results obtained by specifying the properties variable over an element, in other words nodal properties.

C. BOUNDARY CONDITIONS

The load vector represented by equation (9) contains two surface integrals which pertain to boundary conditions described by equation (3). The development of this load vector is shown in Ref. 9.

The usual technique for specification of the boundary conditions is to consider q and h as constants over the applicable element surface. In order to specify convection boundary conditions which vary over an element surface, the flux and heat transfer coefficient have been treated as nodal values in the same manner as nodal material properties. The heat generation term, Q, is also treated in this manner. Therefore, instead of the constant terms in equation (9), the usual shape functions N_i are used with the nodal values to form the variable terms $Q = N_i\, Q_i$, $q = N_i\, q_i$, and $h = N_i\, h_i$. The integration processes proceed as described in Ref. 7, with the addition of these terms.

The surface integrals are computed using a coordinate transformation to allow the use of Gauss Quadrature. This transformation maps the Cartesian coordinates (x,y,z) into

a local coordinate system (ξ,η,ζ). As described by Hanson [10], the unit area vector in the (x,y,z) system is evaluated in the (ξ,η,ζ) system by

$$\hat{n}\, dS_e = \left(\frac{\partial \bar{r}}{\partial \alpha} \times \frac{\partial \bar{r}}{\partial \beta}\right) d\alpha\, d\beta\ , \tag{12}$$

where

$$\bar{r} = x(\xi,\eta,\zeta)\hat{i} + y(\xi,\eta,\zeta)\hat{j} + z(\xi,\eta,\zeta)\hat{k} \tag{13}$$

specifies the surface and α,β represent ξ,η, or ζ.

The vector $V = \frac{\partial \bar{r}}{\partial \alpha} \times \frac{\partial \bar{r}}{\partial \beta}$ is normal to the surface area $d\alpha\, d\beta$ and must be evaluated during the integration process. It is convenient to express equation (12) as

$$\hat{n}\, dS_e = |\bar{V}| \left(\frac{\bar{V}}{|\bar{V}|}\right) d\alpha\, d\beta = |\bar{V}|\hat{\underset{\sim}{n}}\, d\alpha\, d\beta\ , \tag{14}$$

where $\hat{\underset{\sim}{n}}$ is a unit normal vector to surface area $d\alpha\, d\beta$. Therefore, during the integration process, the factor

$$|\bar{V}| = \left[\left(\frac{\partial y}{\partial \alpha}\frac{\partial z}{\partial \beta} - \frac{\partial z}{\partial \alpha}\frac{\partial y}{\partial \beta}\right)^2 + \left(\frac{\partial x}{\partial \alpha}\frac{\partial z}{\partial \beta} - \frac{\partial z}{\partial \alpha}\frac{\partial x}{\partial \beta}\right)^2 + \left(\frac{\partial x}{\partial \alpha}\frac{\partial y}{\partial \beta} - \frac{\partial y}{\partial \alpha}\frac{\partial x}{\partial \beta}\right)^2\right]^{1/2} \tag{15}$$

must be evaluated. If the surface integral is performed over an area on which one of the variables x, y, or z is

constant, the quantity $|\bar{V}|$ becomes identical to the determinant of the Jacobian matrix for the surface transformation. For a general surface $r(x,y,z)$, the complete expressions for $|\bar{V}|$ must be used.

II. STRESS DISTRIBUTION

The finite element technique provides the required methods for the solution of the three dimensional elastic stress analysis problem. Reference 9 describes the derivations for the following equations. A structure can be divided into several elements, each formed by a specified number of nodes. If this structure is acted upon by various forces, the nodal displacements $\{r\}$ are determined by solving the following equation:

$$[K]\{r\} = \{R\} - \{F\}_B - \{F\}_P - \{F\}_{\varepsilon_o} - \{F\}_{\sigma_o} , \qquad (16)$$

where the matrices and vectors are described by

$$[K] = \int_V [B]^T [D] [B] \, dV \qquad (17)$$

$$[F]_B = -\int_V [N]^T \{f_B\} \, dV \qquad (18)$$

$$[F]_P = -\int_S [N]^T \{f_P\} \, dV \qquad (19)$$

$$[F]_{\varepsilon_o} = -\int_V [B]^T [D] \{\varepsilon_o\} \, dV \qquad (20)$$

$$[F]_{\sigma_o} = +\int_V [B]^T \{\sigma_o\} \, dV \qquad (21)$$

and $\{R\}$ is the vector of concentrated nodal loads. The other symbols used in these equations are defined as:

$[N]$	-	vector of shape functions
$[B]$	-	matrix of shape function derivatives
$[D]$	-	elasticity matrix
$\{f_B\}$	-	vector of body forces
$\{f_p\}$	-	vector of surface forces
$\{\varepsilon_o\}$	-	vector of initial strains
$\{\sigma_o\}$	-	vector of initial stresses

Following the solution of equation (16), the nodal displacements $\{r\}$ are used to evaluate the nodal strains.

$$\{\varepsilon\} = [B]\ \{r\} \tag{22}$$

The final step in the analysis is the calculation of the nodal stresses.

$$\{\sigma\} = [D]\ (\{\varepsilon\} - \{\varepsilon_o\}) + \{\sigma_o\} \tag{23}$$

A computer program to make these calculations was written at the Naval Postgraduate School by Leonidas [8]. In that version of the program, the right hand side load vector was allowed to contain only concentrated loads $\{R\}$. In order to permit the presence of thermal stresses and body

forces, several changes were made to allow $\{\varepsilon_o\}$ and $\{f_B\}$ to have non-zero values.

A. GENERATION OF LOAD VECTORS

Temperature differences always generate strains but sometimes stresses may still be zero, for example, when the temperature differences are constant throughout, ε in equation (23) would be equal to ε_o giving zero stress for one material. The presence of a non-uniform temperature distribution in a solid body causes stresses and strains due to non-uniform expansion. In addition, a solid body composed of materials with differing coefficients of linear expansion, λ, will at times experience stresses and strains. This happens when the temperature distributions differ from that of the zero-stress condition.

The forces present in a body during a thermal stress state can be calculated by setting $\varepsilon_o = \lambda\Delta T$, where ΔT is the difference between the actual temperature distribution and the zero-stress temperature distribution. In order to form a consistent load vector, these forces, calculated by equation (20), are added to any other nodal forces present on the right hand side of equation (16).

The body forces, due to some force field, are converted into nodal forces by using equation (17). These are added to the right hand vector of equation (16) to form the total consistent load vector.

B. NODAL MATERIAL PROPERTIES

As discussed in Section IB, the use of nodal material properties is a more convenient method for composite body analysis. In the stress analysis operations, the elastic matrix and the coefficient of linear expansion enter into the calculations. The elastic matrix is composed of the elastic modulus and Poisson's ratio. For the different operations, these properties are evaluated by using $E = N_i\ E_i$, $\mu = N_i\ \mu_i$, and $\lambda = N_i\ \lambda_i$. During the various integration processes these property values are evaluated at each integration step.

The use of nodal properties requires careful consideration as to the effect the shape functions have on the properties within the element. If an element of one material is adjacent to an element of another material, the use of nodal properties can lead to erroneous results. Since the nodes common to the elements can have only one set of property values, one of the elements will not have the expected uniform properties. It would be better to use constant element properties or combine the two elements into one and use variable properties over that element. The amount of property variation between the two elements will also assist the analyst in determining the proper technique.

III. DESCRIPTION OF THE PROGRAMS

Three programs were used to complete the thermal stress analysis. As noted in Sections I and II, the programs of Lew and Leonidas [7,8] were used as a basis to develop the programs for the temperature distribution and elastic stress analysis. The third program was written to process the information generated in the stress analysis. The computer listings for these three programs are provided in Appendices F, G, and H.

A. TEMPERATURE PROGRAM

The main alterations to the temperature program [7] were: (1) the elimination of the second time derivative of the quasi-harmonic equation, (2) the addition of the time integration scheme described in Section IA, and (3) changes to subroutines FORMC, CUBE, and BCOND to allow the specification of nodal material properties and nodal boundary conditions. The version of the program listed in Appendix F allows 278 nodes and a bandwidth of 40. These specifications were varied to adjust the computed core requirements to the particular problem being solved. A problem with 278 nodes and a bandwidth of 40 required approximately 350,000 bytes of storage.

The assumptions made in the solution of a problem using this program are: (1) material properties are constant with time and independent of temperature, (2) directional material

properties must align with the global coordinate axes, and (3) the boundary conditions are constant within a heat transfer phase, but may be altered at the end.

1. Subroutine TIMEHB

As discussed in Section IA, this subroutine performs a time integration using a five point backward difference method. During each heat transfer phase, boundary conditions are constant. Following applications of these boundary conditions to the appropriate matrices and vectors, the trapezoid integration method is used to generate three starting values. Along with the initial temperature distribution, these three starting values provide the necessary information to use the five point difference scheme. The output of temperature distributions is controlled in a manner similar to that used in the original program [7]. These output values are adjusted to the reference temperature supplied in the data deck. During a stress analysis problem, this reference temperature is the zero-stress temperature. Since these temperature distributions are used in a subsequent stress analysis, proper control is provided to punch cards or transfer the information to temporary storage devices. The program in its present form allows up to twenty separate temperature vectors to be stored for future use.

2. Subroutine FORMC, CUBE

Subroutine FORMC was modified to allow nodal properties to be used in the integration process for the matrices

[H] and [C]. The subroutine CUBE performs the cubic integration used in the formation of these matrices and therefore required slight modification.

3. Subroutine BCOND

Subroutine BCOND and its subsidiary routines were modified to accommodate nodal boundary conditions. Subroutine JACOB was altered to evaluate the factor given in equation (15). This factor was used in the surface integration as discussed in Section IC.

4. Data Preparation for the Temperature Program

The data required for a transient solution is comprised of eight groupings of cards. Careful detail must be followed to insure proper formats are used and the correct order of cards maintained. A data check can be performed without any major calculations. The data check normally requires 20-30 seconds, however, the full computer core requirement is needed.

Preparation of the data cards is explained in the initial comment cards of the program listing in Appendix F. A sample data deck for a 32 node, 2 element program is provided in Appendix A. Several aspects of the data preparation must be thoroughly understood.

a. Nodal Properties

The method of input for values which are common to several nodes was provided to reduce the number of input cards. Care must be taken to insure that every node has

properties assigned to it, and that the order of material property cards follow the order of ascending nodal numbers.

b. Output Storage

The number of times stored output is requested must match the number of loads used in the stress analysis program.

c. Steady State Solution

If a steady state problem is being analyzed and no convection or internal generation conditions exist, a blank card must be inserted in place of the heat transfer phase data.

d. Stacking of Problems

If this program is used in conjunction with the elastic stress analysis program, only one problem can be loaded at a time. Therefore, the final comments concerning stacking problems do not apply, and a "STOP" card must be used.

e. Boundary Condition Data

When preparing the boundary condition data, the number of cards with boundary property values (NODECD) is given on the first card of each boundary condition data set. Ensure that NODECD cards are used in that particular set.

5. Job Control Language

The necessary job control language for using the temperature program is listed in Appendix B. The two cards beginning with //GO.FT10F001 and //GO.FT11F001 are

necessary for the storage of the matrices [H] and [C] during the time integration process. They are not required for the steady state solution. The card beginning with //GO.FT07F001 is used to store temperature distributions on temporary files for use in the stress analysis program. If punched output is desired, this card should be replaced by //GO.FT07F001 DD SYSOUT=B

The card beginning with //GO.FT06F001 is necessary to allow large amounts of output from the computer printing devices. If less than 3000 lines of output is anticipated, this card can be removed. These job control cards are designed for use at the W.R. CHURCH COMPUTER CENTER, NAVAL POSTGRADUATE SCHOOL, Monterey, California. Use of the program at another installation will require appropriate modifications.

B. ELASTIC STRESS ANALYSIS PROGRAM

Several changes were required to add a transient thermal stress analysis capability to the stress analysis program [8]. These included: (1) Addition of a consistent load generation subroutine LODGEN, (2) Alteration of the subroutine SOLVE to allow multiple loads, (3) Changes to subroutines INPUT, SORT, STRESS to provide steps for various calculations pertaining to the temperature distributions, (4) The capability to use nodal properties requiring the addition of subroutine NDPROP and numerous alterations to the integration process for the computation of the "stiffness matrix", and (5) The addition of a subroutine PRNSTR to calculate the magnitude of the principal stresses.

The program uses external storage for all information. Therefore, numerous additions and changes were made to the various WRITE and READ statements. A thorough knowledge of the direct-access statements used in Fortran IV is essential to understand the program used in this analysis.

1. Subroutine LODGEN

This subroutine performs calculations used to form the consistent load vectors as described in Section IIA. A lengthy computing time is required for this subroutine due to the number of computations and the many direct access file operations. The number of Gauss points (NGPL) used can be specified independently of the number used for the formation of the "stiffness matrix". The consistent load vector forms the right hand side of equation (16) and must be of sufficient accuracy to insure valid final results. For the calculations made in Part I, four Gauss points were considered sufficient.

2. Subroutine SOLVE

The basic subroutine used to solve equation (15) is contained in Refs. 8 and 11. In order to solve multiple load vectors, changes were incorporated to make necessary computations to all load vectors simultaneously. As described in Refs. 8 and 11, calculations are performed on each block of values in the load vector until all nodes have been processed. The changes which were made performed these calculations to a specified block for all load vectors prior to advancing to the next block. This method was used

to minimize the number of computations and the direct access operations.

3. Subroutine NDPROP

This subroutine calculates the material properties at a node or Gauss point. If the elastic matrix is requested, the various properties are calculated for the position in question, and the matrix is formed. If the coefficient of thermal expansion is requested, it is calculated for the position specified and returned to the calling program. This subroutine uses a separate subroutine, SHAPE, to evaluate the shape function values used in the calculation of the property values.

4. Preparation of Input Data

The preparation of all data cards is explained in the initial comment cards of the program listing, Appendix G. Special care must be given to the details of the input cards.

a. Object Time Formats

The use of object time formats is made throughout the input subroutine. The use of object time formats eliminates the need to recall what format a certain data card is to follow. The object time format technique is covered in most books on the Fortran language.

b. Data Check

A data check can be made by specifying NGPS=0. This requires the full 250,000 bytes of storage, but is completed in less than 20 seconds.

c. External Storage of Load Conditions

If the load conditions of temperature distribution and body forces are prepared by a preprocessing program, these values are passed using files 33 and 34. File 35 for use with nodal displacements was added to the input routine for future development. When the values are passed from a preprocessing program, a time value must be provided at the beginning of each set of load conditions.

5. Job Control Language

Since the stress analysis program stores all information on external devices, the job control language is a very critical part of the program. A very thorough understanding of the specific requirements for a particular computer facility is needed to use this program. All examples and instructions presented here apply to the W.R. CHURCH COMPUTER CENTER AT THE NAVAL POSTGRADUATE SCHOOL, Monterey, California.

External storage operations used in the program are divided into two types. One type is the sequential file READ or WRITE statement. This method of external storage performs the operation on a specified record length in a sequential manner. The second type of operation is the direct access method. In this case, a specified record length is written into or retrieved from a given location within the file. In the first type of operation, no changes to the program language are necessary. The second

type of operation requires changes to the DEFINE FILE statements listed in the control program. These statements will vary according to the size of the problem being analyzed.

Each file used by either a sequential type operation or a direct access operation must be described in the job control language. For the case of the direct access operations, the DEFINE FILE statements and the job control language must agree both in record length and the number of records.

In order to provide a complete summary of all files, Table 5 was prepared. If a particular problem does not require a file, its job control language and any DEFINE FILE statements can be eliminated. Table 6 describes each DEFINE FILE statement and job control card. For each file, a formula is supplied which determines the record length and the number of records which that file will contain. The program listing, Appendix G, and the job control card listing, Appendix D, are examples for use with a thermal stress problem using 278 nodes, 27 elements, and 10 load vectors.

C. POST PROCESSING PROGRAM

The elastic stress analysis program provides a considerable amount of information. For a problem of 200 nodes with 10 load conditions, about 16,000 lines of output are generated. Interpretation of this amount of output is both tedious and time consuming. Usually only a few nodes are

TABLE 5

FILE DEFINITIONS

File Number	-	Definition
7	-	Storage of Total Stiffness Matrix Blocks
8	-	Storage of Element Stiffness Matrices
9	-	Storage of Element Connectivity
10	-	Storage of Nodal Coordinate Vectors
11	-	Storage of Nodal Properties
12	-	Storage of Nodal Stress Vectors
13	-	Storage of Nodal Strain Vectors
14	-	Storage of Concentrated Load Vectors
15	-	Storage of Boundary Condition Vectors
16	-	Storage of Element Material Numbers
17	-	Storage of Reaction Vectors
18	-	Storage of Nodal Temperatures
19	-	Storage of Nodal Coordinates by Element
20	-	Storage of Load Vector Blocks
21	-	Storage of Nodal Displacement Vectors
22	-	Storage of Predefined Boundary Displacements
23	-	Storage of Nodal Properties by Element
24	-	Storage of Nodal Temperatures by Element
25	-	Storage of Nodal Body Forces
26	-	Storage of Nodal Displacements
27	-	Storage of Nodal Body Forces by Element

File Number	-	Definition
28	-	Storage of Nodal Displacements by Element
29	-	Storage of Nodal Force Vector
30	-	Storage of Time Vectors
33	-	Storage of Preprocessed Temperature Distributions
34	-	Storage of Preprocessed Body Force Distributions
35	-	Storage of Preprocessed Nodal Displacement Distributions

TABLE 6

DEFINE FILE Statements and Job Control Cards

General format for these cards:

```
DEFINE FILE XX(NRX,LRWX,U,IXX)
//GO.FTXXF001 DD UNIT=SYSDA,SPACE=(LRBX,(NRX,1)),DISP=(NEW,DELETE)
```

where XX,NRX,LRBX are defined in the table.

FILE	Number of Records	Record length in BYTES	Record length in WORDS
XX	NRX	LRBX	LRWX
7	MMxNNxNREC7	NCOUNTx8/NREC7	NCOUNTx2/NREC7
8	4xNEL	7200	1800
9	NEL	80	20
10	NDPT	24	6
11	NDPT	24	6
12	NDPTxNLOAD	56	14
13	NDPT	56	14
14	NCLD	32	8
15	NPBC	16	4
16	NEL	4	1
17	NPBC	32	8
18	NDPTxNLOAD	8	2
19	NEL	480	120
20	NNxNLOAD	NSx8	NSx2
21	NNxNS/NDF	24	6

XX	NRX	LRBX	LRWX
22	NPBC2	24	6
23	NEL	480	120
24	NLOADxNEL	160	40
25	NLOADxNDPT	24	6
26	NLOADxNDPT	24	6
27	NLOADxNEL	480	120
28	NLOADxNEL	480	120
29	NLOADxNDPT	24	6
30	NLOAD	8	2
33	NLOADxNDPT	8	2
34	NLOADxNDPT	24	6
35	NLOADxNDPT	24	6

Notes:

(a) For a 20 nodal point brick, NREC7 = 4

(b) For a 32 nodal point brick, NREC7 = 6

(c) Variables MM, NN, NS, etc. are defined in Ref. 8

considered at one time and the remainder of the information is reserved for later use.

Through the use of job control language, the important information can be stored on external storage devices and processed by a separate program. A post-processing program was written to analyze the stored information.

1. Program Description

The program listed in Appendix G performs the following operations: (a) information stored on external devices is read for a specified number of nodes; (b) the x,y, and z components of all stress vectors are analyzed for the principal stress values, which are used to determine the maximum and minimum values for a given load condition; (c) Offline and Calcomp plots of stress versus coordinate position and temperature versus coordinate position can be obtained; (d) Offline and Calcomp plots can be obtained for the stress values in polar coordinates.

2. Input Data Preparation

The input data deck has been designed to consist of three or four cards. The preparation of these cards is explained in the initial comments of the program in Appendix H. When more than one problem is processed, care must be taken to limit the number of Calcomp plots requested. All Calcomp plots are performed twice to make the lines dark enough for reproduction by other means.

3. Job Control Language

Since the information is to be passed from the stress analysis program, a few changes to the job control language discussed in Section IIIB are necessary. Files 10, 12, 18, and 30 are changed to the following format:

```
//GO.FTXXF001 DD UNIT=XXXXX,SPACE=(LRBX,(NRX,1)),
// DISP=(NEW,KEEP),DSN=YYYYY,VOLUME=SER=ZZZZZ
```

where XX, LRBX, NRX - defined in Section IIIB

XXXXX - a unit number

YYYYY - a file name, e.g. STRESS

ZZZZZ - a device number

The unit number and device number are obtained from the particular computer facility.

The job control language for the post-processing program must match with that passed by the elastic stress analysis program. Files 8, 9, 10, and 11 are used for the stress values, coordinate values, temperature values, and time values respectively. The job control cards for the post-processor have the same form as those in the elastic stress analysis program. The linkage between the two programs is formed by specifying the same unit number, device number, and file name for the information given by the stress program and received by the post-processor. For example, file 12 in the stress program would be

```
//GO.FT12F001 DD UNIT=2311,SPACE=(LRBX,(NRX,1)),
// DISP=(NEW,KEEP),DSN=STRESS,VOLUME=SER=SYS001
```

and, therefore, File 8 in the post-processor would be

```
//GO.FT08F001 DD UNIT=2311,SPACE=(LRBX,(NRX,1)),
// DISP=(OLD,KEEP),DSN=STRESS,VOLUME=SER=SYS001   .
```

The post processing program also contains DEFINE FILE statements which follow the guidelines in Section IIIB.

Further details on the procedures for job control language can be found in Ref. 12.

D. SYSTEM USAGE OF ALL PROGRAMS

The discussions thus far have pertained to the individual programs, however, the actual production usage of the programs was done in a multistep process. The usage of the programs as a system first requires the formation of the system, then the production run of a problem.

1. System Formation

Appendix D lists the job control language used to create the library of programs. Details of this procedure are provided in Ref. 12.

2. Production Runs

Appendix E lists the job control language for the multistep process of temperature and stress distribution determination. The major change in the methods is the transport of the temperature distribution from the first step to the second step. This is accomplished by the card beginning with //GO.FT07F001 for the first step job control language and the card beginning with //GO.FT33F001 of the second step job control language. These two cards connect the informatin generated and used by the two programs.

Reference 12 contains full details of the job control language used.

IV. RECOMMENDATIONS

As with any endeavor to construct a usable tool for engineering analysis, the job is never done. The following changes are recommended for future development:

A. PROGRAM FOR THE TEMPERATURE DISTRIBUTION

The present version of the temperature distribution is designed for an "in-core" solution technique, which restricts the problem size to within the computer core size. The stress program uses an "out of core" technique and is restricted only by the availability of external storage devices. In order to eliminate the core size restriction, the temperature program requires complete revision to permit the use of external storage devices.

B. INITIAL STRESS DISTRIBUTION

The load generator should be altered to allow a non-zero initial stress distribution.

C. JOB CONTROL LANGUAGE

The amount of job control language allowed at a particular computer facility is limited. For the facility used during this analysis, it appears that no more job control language can be used with this program. The various files used in this program must be combined to reduce the number of job control cards needed.

D. MATERIAL PROPERTIES

In order to more accurately solve the real engineering problems, the capability of temperature dependent properties should be added to both the temperature program and the stress program.

APPENDIX A

SAMPLE DATA DECK FOR TEMPERATURE PROGRAM

```
01234567890123456789012345678901234567890123456789012345678901234567890123456789

1  BASE METAL,TWO ELEMENTS
  32    2   -1    4    2    0    2    1    2    2    0.0
 1 32 0.0033597222222       0.0033597222222   0.0033597222222    0.032025
                1        0.0             0.0              0.60000
                2        0.0             0.0              0.58750
                3        0.0             0.0              0.57500
                4        0.0             0.0              0.56250
                5        0.0             0.0              0.55000
                6        1.50000         0.0              0.60000
                7        1.50000         0.0              0.57500
                8        1.50000         0.0              0.55000
                9        3.00000         0.0              0.60000
               10        3.00000         0.0              0.58750
               11        3.00000         0.0              0.57500
               12        3.00000         0.0              0.56250
               13        3.00000         0.0              0.55000
               14        0.0             0.21213          0.51213
               15        0.0             0.19445          0.49445
               16        0.0             0.17677          0.47677
               17        3.00000         0.21213          0.51213
               18        3.00000         0.19445          0.49445
               19        3.00000         0.17677          0.47677
               20        0.0             0.30000          0.30000
               21        0.0             0.28750          0.30000
               22        0.0             0.27500          0.30000
               23        0.0             0.26250          0.30000
               24        0.0             0.25000          0.30000
               25        1.50000         0.30000          0.30000
               26        1.50000         0.27500          0.30000
               27        1.50000         0.25000          0.30000
               28        3.00000         0.30000          0.30000
               29        3.00000         0.28750          0.30000
               30        3.00000         0.27500          0.30000
               31        3.00000         0.26250          0.30000
               32        3.00000         0.25000          0.30000
    1    32   932.0
    1     1 28 25 20 14  1  6  9 17 29 21  2 10 30 26 22 15  3  7 11 18
    2     1 30 26 22 15  3  7 11 18 31 23  4 12 32 27 24 16  5  8 13 19
 0.5000  1.0000
 150     1    4    0.00000    0.00100    1.00000
```

```
 1    2    3     8
 1    1 -5.769675925
 6    6 -5.769675925
 9    9 -5.769675925
17   17 -5.769675925
28   28 -5.769675925
25   25 -5.769675925
20   20 -5.769675925
14   14 -5.769675925
 1    3    3     8
28   28 0.00289351851
25   25 0.00289351851
20   20 0.00289351851
14   14 0.00289351851
 1    1 0.00289351851
 6    6 0.00289351851
 9    9 0.00289351851
17   17 0.00289351851
 2    2   -3     8
32   32 -2.592592592
27   27 -2.592592592
24   24 -2.592592592
16   16 -2.592592592
 5    5 -2.592592592
 8    8 -2.592592592
13   13 -2.592592592
19   19 -2.592592592
 2    3   -3     8
32   32 0.00289351851
27   27 0.00289351851
24   24 0.00289351851
16   16 0.00289351851
 5    5 0.00289351851
 8    8 0.00289351851
13   13 0.00289351851
19   19 0.00289351851
 3    1    4    1.00000    0.10000    4.00000
 1    2    3     8
 1    1 -5.769675925
 6    6 -5.769675925
 9    9 -5.769675925
17   17 -5.769675925
28   28 -5.769675925
25   25 -5.769675925
20   20 -5.769675925
14   14 -5.769675925
 1    3    3     8
28   28 0.00289351851
```

```
25    25 0.00289351851
20    20 0.00289351851
14    14 0.00289351851
 1     1 0.00289351851
 6     6 0.00289351851
 9     9 0.00289351851
17    17 0.00289351851
 2     2   -3    8
32    32 -2.592592592
27    27 -2.592592592
24    24 -2.592592592
16    16 -2.592592592
 5     5 -2.592592592
 8     8 -2.592592592
13    13 -2.592592592
19    19 -2.592592592
 2     3   -3    8
32    32 0.00289351851
27    27 0.00289351851
24    24 0.00289351851
16    16 0.00289351851
 5     5 0.00289351851
 8     8 0.00289351851
13    13 0.00289351851
19    19 0.00289351851
STOP
```

APPENDIX B

JOB CONTROL CARD LISTING
FOR TEMPERATURE PROGRAM

```
01234567890123456789012345678901234567890123456789012345678901234567890123456789

// (JOBNAME CARD)
// EXEC FORTCLG,REGION.GO=350K
//FORT.SYSIN DD *

      SOURCE CODE OF TEMPERATURE PROGRAM FOLLOWED BY /* CARD

//GO.FTC7F001 DD UNIT=SYSDA,SPACE=(CYL,(2,1)),DSN=&TEMP,DISP=(NEW,PASS)
//GO.FT10F001 DD UNIT=SYSDA,SPACE=(CYL,(2,1)),DISP=(NEW,DELETE)
//GO.FT11F001 DD UNIT=SYSDA,SPACE=(CYL,(2,1)),DISP=(NEW,DELETE)
//GO.SYSIN DD *

      CATA DECK FOLLOWED BY /* CARD
```

APPENDIX C

SAMPLE DATA DECK FOR STRESS PROGRAM

```
01234567890123456789012345678901234567890123456789012345678901234567890123456789

START        2
1  TEST  PROBLEM
    2   32   60   20   -1    0   26    0   32    0   -1    0    0    4    3    5
  ELEMENT CONNECTIVITY
 (10I5,/,11I5)
   28   25   20   14    1    6    9   17   29   21
    2   10   30   26   22   15    3    7   11   18    1
   30   26   22   15    3    7   11   18   31   23
    4   12   32   27   24   16    5    8   13   19    1
  JOINT COORDINATES
 (I16,3F15.5,I5)
               1     0.0            0.0            0.60000
               2     0.0            0.0            0.58750
               3     0.0            0.0            0.57500
               4     0.0            0.0            0.56250
               5     0.0            0.0            0.55000
               6     1.50000        0.0            0.60000
               7     1.50000        0.0            0.57500
               8     1.50000        0.0            0.55000
               9     3.00000        0.0            0.60000
              10     3.00000        0.0            0.58750
              11     3.00000        0.0            0.57500
              12     3.00000        0.0            0.56250
              13     3.00000        0.0            0.55000
              14     0.0            0.21213        0.51213
              15     0.0            0.19445        0.49445
              16     0.0            0.17677        0.47677
              17     3.00000        0.21213        0.51213
              18     3.00000        0.19445        0.49445
              19     3.00000        0.17677        0.47677
              20     0.0            0.30000        0.30000
              21     0.0            0.28750        0.30000
              22     0.0            0.27500        0.30000
              23     0.0            0.26250        0.30000
              24     0.0            0.25000        0.30000
              25     1.50000        0.30000        0.30000
              26     1.50000        0.27500        0.30000
              27     1.50000        0.25000        0.30000
              28     3.00000        0.30000        0.30000
              29     3.00000        0.28750        0.30000
```

```
                30          3.00000        0.27500        0.30000
                31          3.00000        0.26250        0.30000
                32          3.00000        0.25000        0.30000
 MATERIAL PROPERTIES
(2I5,3D15.5)
    1   32    2.48000D 07    3.00000D-01    5.00000D-06
    BOUNDARY CONDITIONS
(4I5)
    1    0    1    0
    2    0    1    0
    3    0    1    0
    4    0    1    0
    5    0    1    0
    6    1    1    0
    7    1    1    0
    8    1    1    0
    9    0    1    0
   10    0    1    0
   11    0    1    0
   12    0    1    0
   13    0    1    0
   20    0    0    1
   21    0    0    1
   22    0    0    1
   23    0    0    1
   24    0    0    1
   25    1    0    1
   26    1    0    1
   27    1    0    1
   28    0    0    1
   29    0    0    1
   30    0    0    1
   31    0    0    1
   32    0    0    1
STOP
```

APPENDIX D

JOB CONTROL CARD LISTING FOR STRESS PROGRAM

```
// (JOBNAME CARD)
// EXEC FORTCLG,REGION.GO=250K
//FORT.SYSIN DD *
```

SOURCE CODE OF STRESS PROGRAM FOLLOWED BY /* CARD

```
//GO.FT07F001 DD UNIT=SYSDA,SPACE=(7200,(0120,1)),DISP=(NEW,DELETE)
//GO.FT08F001 DD UNIT=SYSDA,SPACE=(7200,(0100,1)),DISP=(NEW,DELETE)
//GO.FT09F001 DD UNIT=SYSDA,SPACE=(0080,(0027,1)),DISP=(NEW,DELETE)
//GO.FT10F001 DD UNIT=SYSDA,SPACE=(0024,(0278,1)),DISP=(NEW,DELETE)
//GO.FT11F001 DD UNIT=SYSDA,SPACE=(0024,(0278,1)),DISP=(NEW,DELETE)
//GO.FT12F001 DD UNIT=SYSDA,SPACE=(0056,(2780,1)),DISP=(NEW,DELETE)
//GO.FT13F001 DD UNIT=SYSDA,SPACE=(0056,(0278,1)),DISP=(NEW,DELETE)
//GO.FT14F001 DD UNIT=SYSDA,SPACE=(0032,(0278,1)),DISP=(NEW,DELETE)
//GO.FT15F001 DD UNIT=SYSDA,SPACE=(0016,(0100,1)),DISP=(NEW,DELETE)
//GO.FT16F001 DD UNIT=SYSDA,SPACE=(0004,(0027,1)),DISP=(NEW,DELETE)
//GO.FT17F001 DD UNIT=SYSDA,SPACE=(0032,(0100,1)),DISP=(NEW,DELETE)
//GO.FT18F001 DD UNIT=SYSDA,SPACE=(0008,(2780,1)),DISP=(NEW,DELETE)
//GO.FT19F001 DD UNIT=SYSDA,SPACE=(0480,(0027,1)),DISP=(NEW,DELETE)
//GO.FT20F001 DD UNIT=SYSDA,SPACE=(0480,(0400,1)),DISP=(NEW,DELETE)
//GO.FT21F001 DD UNIT=SYSDA,SPACE=(0024,(0800,1)),DISP=(NEW,DELETE)
//GO.FT23F001 DD UNIT=SYSDA,SPACE=(0480,(0027,1)),DISP=(NEW,DELETE)
//GO.FT24F001 DD UNIT=SYSDA,SPACE=(0160,(0270,1)),DISP=(NEW,DELETE)
//GO.FT29F001 DD UNIT=SYSDA,SPACE=(0024,(2780,1)),DISP=(NEW,DELETE)
//GO.FT30F001 DD UNIT=SYSDA,SPACE=(0008,(0010,1)),DISP=(NEW,DELETE)
//GO.FT33F001 DD UNIT=SYSDA,DSN=&TEMP,DISP=(OLD,DELETE)
//GO.SYSIN DD *
```

DATA DECK FOLLOWED BY /* CARD

APPENDIX E

JOB CONTROL CARD LISTING FOR FORMING PROGRAM LIBRARIES

FIRST FORM THE TEMPERATURE LIBRARY

```
// (JOBNAME CARD)
// EXEC FORTCL,PARM.LINK='NCAL,LET'
//FORT.SYSIN DD *
```

INSERT SOURCE CODE OF TEMPERATURE PROGRAM EXCEPT FOR CONTROL PROGRAM

INSERT /* CARD

```
//LINK.SYSLMOD DD DSNAME=S1312.TRITRNL(INPUT),DISP=(NEW,KEEP),
// UNIT=2321,VOLUME=SER=CEL003,LABEL=RETPD=120,
// SPACE=(CYL,(10,1))
```

INSERT /* CARD

NOW FORM THE STRESS PROGRAM LIBRARY

```
// (JOBNAME CARD)
// EXEC FORTCL,PARM.LINK='NCAL,LET'
//FORT.SYSIN DD *
```

INSERT SOURCE CODE OF THE STRESS PROGRAM EXCEPT FOR CONTROL PROGRAM

INSERT /* CARD

```
//LINK.SYSLMOD DD DSNAME=S1312.TRISOP(MAINE),DISP=(NEW,KEEP),
// UNIT=2321,VOLUME=SER=CEL003,LABEL=RETPD=120,
// SPACE=(CYL,(10,1))
```

INSERT /* CARD

JOB CONTROL CARD LISTING
FOR MULTISTEP PROCESS

```
// (JOBNAME CARD)
//STEP1 EXEC FORTCLG,REGION.GO=350K
//FORT.SYSIN DD *

    INSERT CONTROL PROGRAM FOR TEMPERATURE ROUTINE

  /* CARD

//LINK.SYSLIB DD DISP=SHR,DSNAME=SYS1.FORTLIB
//             DD DISP=SHR,DSNAME=SYS1.MPSLIB
//             DD DISP=SHR,DSNAME=S1312.TRITRNL(INPUT),
// VOLUME=SER=CEL003,UNIT=2321
//GO.FT06F001 DD SPACE=(CYL,(1,1)),SYSOUT=0
//GO.FT07F001 DD UNIT=SYSDA,SPACE=(CYL,(2,1)),DSN=&TEMP,DISP=(NEW,PASS)
//GO.FT10F001 DD UNIT=SYSDA,SPACE=(CYL,(2,1)),DISP=(NEW,DELETE)
//GO.FT11F001 DD UNIT=SYSDA,SPACE=(CYL,(2,1)),DISP=(NEW,DELETE)
//GO.SYSIN DD *

  INSERT TEMPERATURE DATA DECK FOLLOWED BY /* CARD

//STEP2 EXEC FORTCLG,REGION.GO=250K
//FORT.SYSIN DD *

   INSERT CONTROL PROGRAM FOR STRESS ROUTINE

  /* CARD

//LINK.SYSLIB DD DISP=SHR,DSNAME=SYS1.FORTLIB
//             DD DISP=SHR,DSNAME=SYS1.MPSLIB
//             DD DISP=SHR,DSNAME=S1312.TRISOP(MAINE),VOLUME=SER=CEL005,
// UNIT=2321
//GO.FT06F001 DD SPACE=(CYL,(2,1)),SYSOUT=0
//GO.FT07F001 DD UNIT=SYSDA,SPACE=(7200,(0150,1)),DISP=(NEW,DELETE)
//GO.FT08F001 DD UNIT=SYSDA,SPACE=(7200,(0120,1)),DISP=(NEW,DELETE)
//GO.FT09F001 DD UNIT=SYSDA,SPACE=(0080,(0027,1)),DISP=(NEW,DELETE)
//GO.FT10F001 DD UNIT=2311,SPACE=(0024,(0278,1)),DISP=(NEW,KEEP),
// DSN=CUARD3,VOLUME=SER=SYS001
//GO.FT11F001 DD UNIT=SYSDA,SPACE=(0024,(0278,1)),DISP=(NEW,DELETE)
//GO.FT12F001 DD UNIT=2311,SPACE=(0056,(2780,1)),DISP=(NEW,KEEP),
// DSN=STRESS3,VOLUME=SER=SYS001
//GO.FT13F001 DD UNIT=SYSDA,SPACE=(0056,(0278,1)),DISP=(NEW,DELETE)
//GO.FT14F001 DD UNIT=SYSDA,SPACE=(0032,(0278,1)),DISP=(NEW,DELETE)
//GO.FT15F001 DD UNIT=SYSDA,SPACE=(0016,(0120,1)),DISP=(NEW,DELETE)
//GO.FT16F001 DD UNIT=SYSDA,SPACE=(0004,(0027,1)),DISP=(NEW,DELETE)
```

```
//GO.FT17F001 DD UNIT=SYSDA,SPACE=(0032,(0120,1)),DISP=(NEW,DELETE)
//GO.FT18F001 DD UNIT=2311,SPACE=(0008,(2780,1)),DISP=(NEW,KEEP),
// DSN=TEMP3,VOLUME=SER=SYS001
//GO.FT19F001 DD UNIT=SYSDA,SPACE=(0480,(0027,1)),DISP=(NEW,DELETE)
//GO.FT20F001 DD UNIT=SYSDA,SPACE=(0480,(0400,1)),DISP=(NEW,DELETE)
//GO.FT21F001 DD UNIT=SYSDA,SPACE=(0024,(0800,1)),DISP=(NEW,DELETE)
//GO.FT23F001 DD UNIT=SYSDA,SPACE=(0480,(0027,1)),DISP=(NEW,DELETE)
//GO.FT24F001 DD UNIT=SYSDA,SPACE=(0160,(0270,1)),DISP=(NEW,DELETE)
//GO.FT29F001 DD UNIT=SYSDA,SPACE=(0024,(2780,1)),DISP=(NEW,DELETE)
//GO.FT30F001 DD UNIT=2311,SPACE=(0008,(0010,1)),DISP=(NEW,KEEP),
// DSN=TIME3,VOLUME=SER=SYS001
//GO.FT33F001 DD UNIT=SYSDA,DSN=&TEMP,DISP=(OLD,DELETE)
//GO.SYSIN DD *

   INSERT DATA DECK FOR STRESS PROGRAM FOLLOWED BY /* CARD
```

APPENDIX F

COMPUTER LISTING OF TEMPERATURE PROGRAM

```
C***********************************************************************CNTP0010
C                                                                       CNTP0020
C     THIS PROGRAM SOLVES THE TRANSIENT HEAT EQUATION IN THREE          CNTP0030
C     DIMENSIONS. IN ADDITION THE STEADY STATE SOLUTION CAN BE          CNTP0040
C     OBTAINED DIRECTLY.  THIS PROGRAM IS BASED ON A PROGRAM ORIGINALLY CNTP0050
C     WRITTEN BY LCDR. LEW, NAVAL POSTGRADUATE SCHOOL,1972. PRESENT     CNTP0051
C     VERSION WAS WRITTEN BY LT. BUBECK,NAVAL POSTGRADUATE SCHOOL,1975. CNTP0052
C                                                                       CNTP0060
C***********************************************************************CNTP0070
C                                                                       CNTP0080
C                                                                       CNTP0090
C     RESTRICTIONS:                                                     CNTP0100
C                                                                       CNTP0110
C        (1)  PROPERTIES ARE CONSTANT                                   CNTP0120
C        (2)  STATIONARY BOUNDARIES AND ELEMENTS ARE REQUIRED.          CNTP0130
C        (3)  ANISOTROPIC MATERIALS MUST ALIGN WITH AXES.               CNTP0140
C                                                                       CNTP0150
C                                                                       CNTP0160
C     INPUT DATA DECK                                                   CNTP0170
C                                                                       CNTP0180
C        (1) TITLE CARD (10A8) - ANY TITLE IN COLUMNS 2-80; A NUMERAL   CNTP0190
C            "1" PLACED IN COLUMN 1 STARTS PRINTOUT ON A NEW PAGE.      CNTP0200
C                                                                       CNTP0210
C        (2) PROBLEM CHARACTERISTICS (10I5,F20.0)                       CNTP0220
C                                                                       CNTP0230
C            NNODE....NUMBER OF NODES...MAXIMUM OF 278, WITH A MAXIMUM  CNTP0240
C                                       BAND WIDTH OF 40                CNTP0250
C            NEL......NUMBER OF ELEMENTS...MAXIMUM OF 48                CNTP0260
C            NMAT.....NUMBER OF MATERIALS...MAXIMUM OF 10 ELEMENT       CNTP0270
C                                           MATERIALS...NMAT=-1 IF NODALCNTP0280
C                                           MATERIALS                   CNTP0290
C            NGP......NUMBER OF GAUSS POINTS...FROM 2-->6,DEFAULT=3,    CNTP0300
C                     IF NGP = 0 , A DATA CHECK IS PERFORMED.           CNTP0310
C            IGO......ELEMENT TYPE...1 - LINEAR;2 - QUADRATIC;3 - CUBIC CNTP0320
C            NNBC.....NUMBER OF SPECIFIED NODAL VALUES...MAXIMUM OF 150 CNTP0330
C            IPROB....PROBLEM TYPE...1 - STEADY STATE; 2 - TRANSIENT    CNTP0340
C            KC.......COORDINATE SYSTEM...1 - RECTANGULAR;              CNTP0350
C                                         2 - CYLINDRICAL;              CNTP0360
C                                         3 - SPHERICAL                 CNTP0370
C            IPUN.....PUNCHED OUTPUT INDICATOR...0 - NO PUNCH; A POSITIVCNTP0380
C                     INTEGER SPECIFIES THE NUMBER OF SETS OF VALUES TO CNTP0390
C                     BE PUNCHED OR STORED.                             CNTP0400
C            NTSC.....NUMBER OF HEAT TRANSFER PHASES...EACH PHASE HAS   CNTP0410
```

```
C             A NEW SET OF BOUNDARY CONDTIONS.                           CNTP0420
C       PHIREF...REFERENCE VALUE FOR PHI.                                CNTP0430
C                                                                        CNTP0440
C   (3) MATERIAL PROPERTIES - IF ELEMENTAL PROPERTIES, NMAT CARDS,      CNTP0450
C                             IF NODAL PROPERTIES, NUMBER OF CARDS      CNTP0460
C                             IS VARIABLF.                              CNTP0470
C       FOR ELEMENTAL PROPERTIES (I5,4F15.0)                            CNTP0480
C       ONE INTEGER FOR MATERIAL NUMBER, FOUR FLOATING POINT            CNTP0490
C       NUMBERS FOR KX,KY,KZ,AND RHO*C.                                 CNTP0500
C                                                                        CNTP0510
C       FOR NODAL PROPERTIES (2I3,4F18.0)                               CNTP0520
C       CASE I:  IF ALL NODES(CONSECUTIVE) BETWEEN NODE "N" AND         CNTP0530
C       NODE "M" ARE OF SAME MATERIAL, THE TWO INTEGERS ARE "N" ANDCNTP0540
C       "M" RESPECTIVELY, AND THE FOUR FLOATING POINT NUMBERS ARE       CNTP0550
C       PROPERTIES.                                                     CNTP0560
C       CASE II: IF A NODE "N" HAS UNIQUE PROPERTIES, BOTH              CNTP0570
C       INTEGERS ARE "N" AND PROPERTIES FOLLOW.                         CNTP0580
C                                                                        CNTP0590
C   (4) NODE CARDS; TWO SETS OF NNODE CARDS PER PROBLEM                 CNTP0600
C       (DATA ASSUMED ZERO IF NOT SPECIFIED)                            CNTP0610
C                                                                        CNTP0620
C       SET  I:                                                          CNTP0630
C        COLS  1-10 NODE NUMBER                                          CNTP0640
C        COLS 11-30 X COORDINATE OR RADIUS                               CNTP0650
C        COLS 31-50 Y COORDINATE OR ANGLE IN DEGREES RELATIVE TO         CNTP0660
C                   X AXIS (+ CCW)                                       CNTP0670
C        COLS 51-70 Z COORDINATE OR ANGLE IN DEGREES RELATIVE TO         CNTP0680
C                   Z AXIS (+ CW)                                        CNTP0690
C                                                                        CNTP0700
C       SET II: (NOT REQUIRED FOR THE STEADY STATE PROBLEM)              CNTP0710
C        CASE I:  IF ALL NODES(CONSECUTIVE) BETWEEN NODE "N" AND         CNTP0720
C        NODE "M" ARE AT SAME INITIAL TEMPERATURE, THE TWO INTEGERSCNTP0730
C        ARE "N" AND "M" RESPECTIVELY, AND THE FLOATING POINT            CNTP0740
C        NUMBER IS THE TEMPERATURE..                                     CNTP0750
C        CASE II: IF A NODE "N" HAS A UNIQUE INITIAL TEMPERATURE,        CNTP0760
C        BOTH INTEGERS ARE "N" AND THE TEMPERATURE FOLLOWS.              CNTP0770
C                                                                        CNTP0780
C   (5) NODE BOUNDARY VALUES; NNBC CARDS PER PROBLEM                     CNTP0790
C        INPUT IN ORDER OF ACENDING NODE NUMBERS                         CNTP0800
C       (OMIT IF DATA NOT REQUIRED)                                      CNTP0810
C         COLS  1-10  JOINT NUMBER                                       CNTP0820
C         COLS 11-30  SPECIFIED NODAL VALUE                              CNTP0830
C                                                                        CNTP0840
C   (6) CONNECTIVITY DATA, NEL CARDS PER PROBLEM (2I5,32I3)             CNTP0850
C         COLS  1-5  ELEMENT NUMBER                                      CNTP0860
C         COLS  6-10 MATERIAL IDENTIFICATION NUMBER                      CNTP0870
C         COLS 11-107 CONNECTIVITY, CCW FROM UPPER RIGHT                 CNTP0880
C                     LOOKING INTO Y-Z PLANE FROM (+) X AXIS             CNTP0890
```

```
C                   NODES  1-23 ON CARD I (2I5,23I3)                        CNTP0900
C                   NODES 24-32 ON CARD II (9I3)                            CNTP0910
C                                                                           CNTP0920
C        (7) TIMES FOR PUNCHED OR STORED OUTPUT.  UP TO 20 SEPARATE         CNTP0930
C            TIMES MAY BE REQUESTED IN A FORMAT(10F8.0,/,10F8.0). VALUESCNTP0940
C            ARE PUNCHED OR STORED DEPENDING ON THE JOB CONTROL LANGUAGECNTP0950
C            USING AN OUTPUT FORMAT OF  (I5,F20.0).                         CNTP0960
C                                                                           CNTP0970
C     HEAT TRANSFER PHASE DATA                                              CNTP0980
C                                                                           CNTP0990
C     FOR EACH PHASE, THE REMAINING DATA MUST BE SUPPLIED. IF STEADY        CNTP1000
C     STATE PROBLEM ONLY ONE SET IS USED.                                   CNTP1010
C                                                                           CNTP1020
C                                                                           CNTP1030
C     TRANSIENT PROBLEM DATA ( SUPPLY ONE BLANK CARD FOR STEADY STATE       CNTP1040
C     PROBLEM WHICH HAS NELBC = 0), OTHERWISE (3I5,3F10.0)                  CNTP1050
C        NUMBER OF INITIAL STEPS(ISTP)                                      CNTP1060
C        NUMBER OF STEPS BETWEEN PRINTINGS(IPRT)                            CNTP1070
C        NUMBER OF BOUNDARY CONDITIONS(NELBC), MAXIMUM OF 48                CNTP1080
C        BEGIN TIME FOR THIS PHASE(TIM)                                     CNTP1090
C        INITIAL STEPSIZE FOR THIS PHASE(DT)                                CNTP1100
C        END TIME FOR THIS PHASE(ENDT)                                      CNTP1110
C                                                                           CNTP1120
C     BOUNDARY CONDITION DATA: NELBC SETS PER PHASE                         CNTP1130
C        EACH SET PERTAINS TO A PARTICULAR CONDITION ON ONE ELEMENT         CNTP1140
C        SURFACE. A SET CONTAINS A FIRST CARD PLUS A VARIABLE NUMBER        CNTP1150
C        OF CARDS.                                                          CNTP1160
C                                                                           CNTP1170
C        FIRST CARD: (4I5), ELEMENT NUMBER, BOUNDARY CONDTION TYPE(KBT),CNTP1180
C        SURFACE APPLICABLE(KBS),NUMBER OF CARDS SPECIFYING PROPERTY        CNTP1190
C        VALUE(NODECD); WHERE                                               CNTP1200
C                 BOUNDARY CONDITION TYPE(KBT)                              CNTP1210
C                   1 = VOLUME INTEGRAL OF LOADING                          CNTP1220
C                   2 = SURFACE INTEGRAL OF LOADING                         CNTP1230
C                   3 = SURFACE LOSS COEFFICIENT                            CNTP1240
C                 SURFACE APPLICABLE (KBS)                                  CNTP1250
C                   0 = VOLUME INTEGRAL (USE WITH KBT = 1 ONLY)             CNTP1260
C                   1 = SURFACE INTEGRAL OVER XI = +/- 1 FACE               CNTP1270
C                   2 = SURFACE INTEGRAL OVER ETA = +/- 1 FACE              CNTP1280
C                   3 = SURFACE INTEGRAL OVER ZETA = +/- 1 FACE             CNTP1290
C                   *** KBS = 1,2 OR 3 MUST BE SIGNED TO INDICATE ***       CNTP1300
C                   *** PLUS (+) OR MINUS (-) FACE                ***       CNTP1310
C                                                                           CNTP1320
C        NODECD CARDS(VARIABLE),(2I5,1G20.8)                                CNTP1330
C        IF ALL NODES(CONSECUTIVE NUMBERING) ON PARTICULAR ELEMENT          CNTP1340
C        SURFACE BETWEEN NODE "N" AND NODE "M" HAVE SAME BOUNDARY           CNTP1350
C        CONDITION PROPERTY VALUE, THE TWO INTEGERS ARE "N" AND "M",        CNTP1360
C        AND PROPERTY VALUE FOLLOWS.                                        CNTP1370
```

```
C         IF A NODE "N" HAS UNIQUE PROPERTY VALUES, BOTH INTEGERS ARE "N"CNTP1380
C         AND THE PROPERTY VALUE FOLLOWS.                                 CNTP1390
C                                                                         CNTP1400
C      STACK PROBLEMS TO BE SOLVED.  TO TERMINATE RUN ADD DATA CARD       CNTP1410
C      WITH "STOP" PUNCHED IN COLS 1-4                                    CNTP1420
C                                                                         CNTP1430
C      ******************************************************************CNTP1440
C      *                                                                *CNTP1450
C      *                         CONTROL PROGRAM                        *CNTP1460
C      *                                                                *CNTP1470
C      ******************************************************************CNTP1480
       IMPLICIT REAL*8(A-H,O-Z)                                           CNTP1490
       COMMON /INT/ IGO,NBAND,NNODE,NMAT,NEL,NPEL,NGP,NELBC,NNBC,IPROB,KCCNTP1500
      1,NTSC,IPUN,IPRT,ISTP                                               CNTP1510
  100 CALL INPUT(1)                                                       CNTP1520
      IF(NGP.EQ.0) GO TO 100                                              CNTP1530
      CALL MERGE(1)                                                       CNTP1540
      IF(IPROB.NE.1) GO TO 150                                            CNTP1550
      CALL INPUT(2)                                                       CNTP1560
      IF(NELBC.NE.0) CALL BCOND                                           CNTP1570
      CALL STEADY                                                         CNTP1580
      GO TO 100                                                           CNTP1590
  150 CALL MERGE(2)                                                       CNTP1600
      CALL TIMEHB                                                         CNTP1610
      GO TO 100                                                           CNTP1620
      END                                                                 CNTP1630
C
C************************************************************************
C************************************************************************
C
      SUBROUTINE INPUT (IFLAG)                                            INP00010
C
C     THIS SUBROUTINE READS ALL REQUIRED DATA
C
      IMPLICIT REAL*8(A-H,O-Z)                                            INP00020
      COMMON /MTRX/ CORD(32,3),ENN(32,32),DCOL(32,3),COL(32),AJ(3,3),AJIINP00030
     1(3,3),ENW(32,32),FE(32),PROP(278,4),COARD(278,3),F(278),PHI(278),PINP00040
     2HIDOT(278),PHIWRK(278),PHIDDT(278),BGH(278,40),BGC(278,40),PROB(27INP00050
     38,3),PUNCH(20)                                                      INP00060
      COMMON /IMTRX/ NCON(48,33),KELB(48,3),KNODE(278)                    INP00070
      COMMON /FLP/ DTJ,AKX,AKY,AKZ,FACT,PHIREF,APZRO,ABIGN,TIM,DT,ENDT    INP00080
      COMMON /INT/ IGO,NBAND,NNODE,NMAT,NEL,NPEL,NGP,NELBC,NNBC,IPROB,KCINP00090
     1,NTSC,IPUN,IPRT,ISTP                                                INP00100
      DIMENSION TITLE(10)                                                 INP00110
      DATA CHK/'STOP    '/                                                INP00120
      IF (IFLAG.EQ.2) GO TO 270                                           INP00130
      READ (5,360) TITLE                                                  INP00140
      IF (TITLE(1).EQ.CHK) STOP                                           INP00150
```

```
      WRITE (6,360) TITLE                                               INP00160
C
C     READ PROBLEM CHARACTERISTICS
C
      READ (5,370) NNODE,NEL,NMAT,NGP,IGO,NNBC,IPROB,KC,IPUN,NTSC,PHIREFINP00170
      WRITE (6,380) NNODE,NEL,NMAT,NGP,NNBC,IPROB,PHIREF                INP00180
      NPEL = 12*(IGO-1)+8                                               INP00190
      WRITE (6,390) IGO,NPEL                                            INP00200
      IF (IPUN.GE.1) WRITE (6,400)                                      INP00210
C
C     ZERO STORAGE AREAS
C
      NP1 = NPEL+1                                                      INP00220
      DO 10 J=1,NP1                                                     INP00230
      DO 10 I=1,NEL                                                     INP00240
   10 NCON(I,J) = 0                                                     INP00250
      DO 20 I=1,NNODE                                                   INP00260
      F(I) = 0.0D0                                                      INP00270
      PHI(I) = 0.0D0                                                    INP00280
      PHIDOT(I) = 0.0D0                                                 INP00290
      PHIWRK(I) = 0.0D0                                                 INP00300
      DO 20 J=1,3                                                       INP00310
   20 CCARD(I,J) = 0.0D0                                                INP00320
      DO 30 J=1,4                                                       INP00330
      DO 30 I=1,NNODE                                                   INP00340
   30 PROP(I,J) = 0.0D0                                                 INP00350
      DO 40 I=1,48                                                      INP00360
      DO 40 J=1,2                                                       INP00370
   40 KELB(I,J) = 0                                                     INP00380
      DO 50 J=1,3                                                       INP00390
      DO 50 I=1,NNODE                                                   INP00400
   50 PROB(I,J) = 0.0D0                                                 INP00410
C
C     READ MATERIAL PROPERTIES
C
      IF (NMAT.LT.0) GO TO 80                                           INP00420
      IF (NMAT.EQ.0) GO TO 130                                          INP00430
      DO 60 I=1,NMAT                                                    INP00440
   60 READ (5,410) ID,(PROP(ID,J),J=1,4)                                INP00450
      WRITE (6,420)                                                     INP00460
      DO 70 I=1,NMAT                                                    INP00470
   70 WRITE (6,430) I,(PROP(I,J),J=1,4)                                 INP00480
      GO TO 130                                                         INP00490
   80 INODE = 1                                                         INP00500
   90 IF (INODE.EQ.NNODE) GO TO 110                                     INP00510
      READ (5,440) INNODE,INODE,PROP1,PROP2,PROP3,PROP4                 INP00520
      DO 100 I=INNODE,INODE                                             INP00530
      PROP(I,1) = PROP1                                                 INP00540
```

```
      PROP(I,2) = PROP2                                                 INP00550
      PROP(I,3) = PROP3                                                 INP00560
  100 PROP(I,4) = PROP4                                                 INP00570
      GO TO 90                                                          INP00580
  110 WRITE (6,450)                                                     INP00590
      DO 120 I=1,NNODE                                                  INP00600
  120 WRITE (6,460) I,(PROP(I,J),J=1,4)                                 INP00610
C
C     READ JOINT COORDINATES, INITIAL NODAL VALUES, ANY FIXED BOUNDARY
C     VALUES.  CONVERT IF NECESSARY
C
  130 DO 140 I=1,NNODE                                                  INP00620
  140 READ (5,470) IJT,(COARD(IJT,J),J=1,3)                             INP00630
      IF (IPROB.LE.1) GO TO 170                                         INP00640
      I = 1                                                             INP00650
  150 IF (I.EQ.NNODE) GO TO 170                                         INP00660
      READ (5,480) IJT1,IJT2,PHIVAL                                     INP00670
      DO 160 I=IJT1,IJT2                                                INP00680
  160 PHI(I) = PHIVAL                                                   INP00690
      GO TO 150                                                         INP00700
  170 IF (NNBC.LE.0) GO TO 200                                          INP00710
      DO 180 I=1,NNBC                                                   INP00720
  180 READ (5,490) KNODE(I),PHIWRK(KNODE(I))                            INP00730
      WRITE (6,500)                                                     INP00740
      DO 190 I=1,NNBC                                                   INP00750
      ID = KNODE(I)                                                     INP00760
  190 WRITE (6,510) ID,PHIWRK(ID)                                       INP00770
  200 WRITE (6,520)                                                     INP00780
      DO 210 I=1,NNODE                                                  INP00790
  210 WRITE (6,530) I,(COARD(I,J),J=1,3),PHI(I)                         INP00800
      IF(KC.GT.1) CALL MARK(COARD,KC,NNODE)                             INP00801
C
C     READ CONNECTIVITY, COMPUTE HALF BAND WIDTH
C
      DO 220 I=1,NEL                                                    INP00810
  220 READ (5,540) IEL,(NCON(IEL,J),J=1,NP1)                            INP00820
      WRITE (6,550)                                                     INP00830
      DO 230 I=1,NEL                                                    INP00840
  230 WRITE (6,560) I,(NCON(I,J),J=1,NP1)                               INP00850
      NBAND = 0                                                         INP00860
      DO 250 I=1,NEL                                                    INP00870
      DO 240 J=2,NPEL                                                   INP00880
      JK = J+1                                                          INP00890
      DO 240 K=JK,NP1                                                   INP00900
  240 NBAND = MAX0(NBAND,IABS(NCON(I,J)-NCON(I,K)))                     INP00910
  250 CONTINUE                                                          INP00920
      NBAND = NBAND+1                                                   INP00930
      WRITE (6,570) NBAND                                               INP00940
```

```
C
C     READ TIMES FOR PUNCHED OR STORED OUTPUT
C
      IF (IPUN.EQ.0) GO TO 260                                          INP00950
      IF (IPROB.EQ.1) GO TO 260                                         INP00960
      READ (5,580) (PUNCH(I),I=1,IPUN)                                  INP00970
      WRITE (6,590) (PUNCH(I),I=1,IPUN)                                 INP00980
  260 IF (NGP.NE.0) RETURN                                              INP00990
C
C     READ PHASE DATA AND BOUNDARY CONDITIONS
C
      IHTC = 1                                                          INP01000
  270 READ (5,600) ISTP,IPRT,NELBC,TIM,DT,ENDT                          INP01010
      WRITE (6,610)                                                     INP01020
      WRITE (6,620) ISTP,IPRT,NELBC,TIM,DT,ENDT                         INP01030
      IF (NELBC.EQ.0) GO TO 350                                         INP01040
      WRITE (6,650)                                                     INP01050
      DO 340 I=1,NELBC                                                  INP01060
      READ (5,640) (KELB(I,J),J=1,3),NODECD                             INP01070
      IEL = KELB(I,1)                                                   INP01080
      NBCTYP = KELB(I,2)                                                INP01090
      GO TO (280,300,320), NBCTYP                                       INP01100
  280 DO 290 II=1,NODECD                                                INP01110
      READ (5,630) IJT1,IJT2,PROPB                                      INP01120
      DO 290 IJT=IJT1,IJT2                                              INP01130
      PROB(IJT,1) = PROPB                                               INP01140
  290 WRITE (6,660) IEL,IJT,NBCTYP,KELB(I,3),PROB(IJT,1)                INP01150
      GO TO 340                                                         INP01160
  300 DO 310 II=1,NODECD                                                INP01170
      READ (5,630) IJT1,IJT2,PROPB                                      INP01180
      DO 310 IJT=IJT1,IJT2                                              INP01190
      PROB(IJT,2) = PROPB                                               INP01200
  310 WRITE (6,660) IEL,IJT,NBCTYP,KELB(I,3),PROB(IJT,2)                INP01210
      GO TO 340                                                         INP01220
  320 DO 330 II=1,NODECD                                                INP01230
      READ (5,630) IJT1,IJT2,PROPB                                      INP01240
      DO 330 IJT=IJT1,IJT2                                              INP01250
      PROB(IJT,3) = PROPB                                               INP01260
  330 WRITE (6,660) IEL,IJT,NBCTYP,KELB(I,3),PROB(IJT,3)                INP01270
  340 CONTINUE                                                          INP01280
  350 IF (NGP.NE.0) RETURN                                              INP01290
      IF (IHTC.EQ.NTSC) RETURN                                          INP01300
      IHTC = IHTC+1                                                     INP01310
      GO TO 270                                                         INP01320
  360 FORMAT (10A8)                                                     INP01330
  370 FORMAT (10I5,F20.0)                                               INP01340
  380 FORMAT (///,11X,' NUMBER OF NODES = ',I5,//,11X,' NUMBER OF ELEMENINP01350
     1TS = ',I5,//,11X,' NUMBER OF MATERIALS = ',I5,//,11X,' NUMBER OF GINP01360
```

```
     2ALSS POINTS = ',I5,//,11X,' NUMBER OF NODES WITH BOUNDARY VALUE = INP01370
     3',I5,//,11X,' PROBLEM TYPE = ',I5,//,11X,' PHI(REFERENCE) = ',1PG2INP01380
     44.16)                                                             INP01390
  390 FORMAT (/,11X,' TYPE',I3,'  ELEMENTS,',I5,'   POINTS PER ELEMENT')INP01400
  400 FORMAT (//,11X,' PUNCHED OR STORED OUTPUT OF PHI AND PHIDOT REQUESINP01410
     1TED')                                                             INP01420
  410 FORMAT (I5,4F15.0)                                                INP01430
  420 FORMAT (///,32X,' MATERIAL PROPERTIES',///,'  MTL# ',7X,' KX',18X,INP01440
     1 ' KY',18X,' KZ',15X,'DENSITY',///)                               INP01450
  430 FORMAT (I5,1P4G20.12,/)                                           INP01460
  440 FORMAT (2I3,4F18.0)                                               INP01470
  450 FORMAT (///,32X,' MATERIAL PROPERTIES',///,'  IJT# ',7X,' KX',18X,INP01480
     1'KY',18X,'KZ',15X,'VISCOSITY',///)                                INP01490
  460 FORMAT (I5,1P4G20.12)                                             INP01500
  470 FORMAT (I16,3F15.0)                                               INP01510
  480 FORMAT (2I5,F20.0)                                                INP01520
  490 FORMAT (I10,F20.0)                                                INP01530
  500 FORMAT (///,21X,' BOUNDARY CONDITION DATA',//,'  NODE',18X,'VALUE'INP01540
     1,///)                                                             INP01550
  510 FORMAT (I6,5X,1PG24.16)                                           INP01560
  520 FORMAT (///,20X,'NODAL COORDINATES AND INITIAL PHI VALUES',///,' NINP01570
     1ODE NO.',10X,'X COORDINATE',12X,'Y COORDINATE',12X,'Z COORDINATE',INP01580
     212X,'PHI INITIAL',///)                                            INP01590
  530 FORMAT (I5,5X,1P5G24.16)                                          INP01600
  540 FORMAT (2I5,23I3,/,9I3)                                           INP01610
  550 FORMAT (//,21X,' CONNECTIVITY MATRIX',///,2X,' EL# MTL#',//)      INP01620
  560 FORMAT (/,2I5,5X,20I4,/,15X,12I4)                                 INP01630
  570 FORMAT (///,11X,' SYSTEM HALF BAND WIDTH = ',I5,///)              INP01640
  580 FORMAT (10F8.4)                                                   INP01650
  590 FORMAT ('0',11X,'DATA WILL BE PUNCHED OR STORED FOR THE FOLLOWING INP01660
     1TIMES : ',//,11X,10F8.4,/,11X,10F8.4)                             INP01670
  600 FORMAT (3I5,3F10.5)                                               INP01680
  610 FORMAT ('1',20X,'DATA FOR THE NEXT HEAT TRANSFER PHASE',///)      INP01690
  620 FORMAT ('0',09X,'NUMBER OF INITIAL STEPS = ',I5,//,10X,  'NUMBER OINP01700
     1F STEPS BETWEEN PRINTINGS = ',I5,//,10X,'NUMBER OF BOUNDARY CONDITINP01710
     2IONS = ',I5,//,10X,'BEGIN TIME FOR THIS PHASE = ',F10.5,//,10X,'ININP01720
     3ITIAL STEP SIZE FOR THIS PHASE = ',F10.5,//,10X,'END TIME FOR THISINP01730
     4 PHASE = ',F10.5)                                                 INP01740
  630 FORMAT (2I5,1G20.8)                                               INP01750
  640 FORMAT (4I5)                                                      INP01760
  650 FORMAT ('0',21X,' BOUNDARY CONDITION DATA',///,'  EL # JT # BC # SINP01770
     1URF     PROPERTY VALUE   ',///)                                   INP01780
  660 FORMAT (' ',4I5,1PG24.16)                                         INP01790
      END                                                               INP01800
C
C***********************************************************************
C***********************************************************************
C
```

```
      SUBROUTINE MERGE (IMTRX)                                          MER00010
C
C     THIS SUBROUTINE FORMS "H" AND "C" MATRICES
C
      IMPLICIT REAL*8(A-H,O-Z)                                          MER00020
      COMMON /MTRX/ CORD(32,3),ENN(32,32),DCOL(32,3),COL(32),AJ(3,3),AJIMER00030
     1(3,3),ENW(32,32),FE(32),PROP(278,4),COARD(278,3),F(278),PHI(278),PMER00040
     2HIDOT(278),PHIWRK(278),PHIDDT(278),BGH(278,40),BGC(278,40),PROB(27MER00050
     38,3),PUNCH(20)                                                    MER00060
      COMMON /IMTRX/ NCON(48,33),KELB(48,3),KNODE(278)                  MER00070
      COMMON /FLP/ DTJ,AKX,AKY,AKZ,FACT,PHIREF,APZRO,ABIGN,TIM,DT,ENDT  MER00080
      COMMON /INT/ IGO,NBAND,NNODE,NMAT,NEL,NPEL,NGP,NELBC,NNBC,IPROB,KCMER00090
     1,NTSC,IPUN,IPRT,ISTP                                              MER00100
C
C     ZERO MASTER AND ELEMENT MATRICES
C
      IF (IMTRX.EQ.2) GO TO 20                                          MER00110
      DO 10 J=1,NBAND                                                   MER00120
      DO 10 I=1,NNODE                                                   MER00130
   10 BGH(I,J) = 0.0D0                                                  MER00140
      GO TO 40                                                          MER00150
   20 DO 30 J=1,NBAND                                                   MER00160
      DO 30 I=1,NNODE                                                   MER00170
   30 BGC(I,J) = 0.0D0                                                  MER00180
   40 DO 140 I=1,NEL                                                    MER00190
      DO 50 JJ=1,3                                                      MER00200
      DO 50 II=1,NPEL                                                   MER00210
   50 CORD(II,JJ) = 0.0D0                                               MER00220
      DO 60 II=1,NPEL                                                   MER00230
      DO 60 JJ=1,NPEL                                                   MER00240
   60 ENW(JJ,II) = 0.0D0                                                MER00250
      DO 70 J=1,NPEL                                                    MER00260
      INDX = NCON(I,J+1)                                                MER00270
      DO 70 K=1,3                                                       MER00280
   70 CORD(J,K) = COARD(INDX,K)                                         MER00290
      IF (IMTRX.EQ.2) GO TO 90                                          MER00300
C
C     FORM ELEMENTAL "H" MATRIX
C
      IF (NMAT.LT.0) GO TO 80                                           MER00310
      IMAT = NCON(I,1)                                                  MER00320
      AKX = PROP(IMAT,1)                                                MER00330
      AKY = PROP(IMAT,2)                                                MER00340
      AKZ = PROP(IMAT,3)                                                MER00350
   80 CALL CUBE (NPEL,NGP,1,I,NMAT)                                     MER00360
      GO TO 120                                                         MER00370
C
C     FORM ELEMENTAL "C" MATRIX
```

```
C
   90 IF (NMAT.LT.0) GO TO 100                                          MER00380
      IMAT = NCON(I,1)                                                  MER00390
      FACT = PROP(IMAT,4)                                               MER00400
  100 CALL CUBE (NPEL,NGP,2,I,NMAT)                                     MER00410
      DO 110 II=1,NPEL                                                  MER00420
      IDX = NCON(I,II+1)                                                MER00430
      DO 110 JJ=1,NPEL                                                  MER00440
      IDY = NCON(I,JJ+1)-IDX+1                                          MER00450
      IF (IDY.LT.1) GO TO 110                                           MER00460
      BGC(IDX,IDY) = BGC(IDX,IDY)+ENW(II,JJ)                            MER00470
  110 CONTINUE                                                          MER00480
      GO TO 140                                                         MER00490
C
C     ASSEMBLE APPROPRIATE MASTER MATRIX
C
  120 DO 130 II=1,NPEL                                                  MER00500
      IDX = NCON(I,II+1)                                                MER00510
      DO 130 JJ=1,NPEL                                                  MER00520
      IDY = NCON(I,JJ+1)-IDX+1                                          MER00530
      IF (IDY.LT.1) GO TO 130                                           MER00540
      BGH(IDX,IDY) = BGH(IDX,IDY)+ENW(II,JJ)                            MER00550
  130 CONTINUE                                                          MER00560
  140 CONTINUE                                                          MER00570
      RETURN                                                            MER00580
      END                                                               MER00590
C
C***********************************************************************
C***********************************************************************
C
      SUBROUTINE STEADY                                                 STY00010
C
C     THIS SUBROUTINE SOLVES THE STEADY STATE PROBLEM
C
      IMPLICIT REAL*8(A-H,O-Z)                                          STY00020
      COMMON /MTRX/ CORD(32,3),ENN(32,32),DCOL(32,3),COL(32),AJ(3,3),AJISTY00030
     1(3,3),ENW(32,32),FE(32),PROP(278,4),COARD(278,3),F(278),PHI(278),PSTY00040
     2HIDOT(278),PHIWRK(278),PHIDDT(278),BGH(278,40),BGC(278,40),PROB(27STY00050
     38,3),PUNCH(20)                                                    STY00060
      COMMON /IMTRX/ NCON(48,33),KELB(48,3),KNODE(278)                  STY00070
      COMMON /FLP/ DTJ,AKX,AKY,AKZ,FACT,PHIREF,APZRO,ABIGN,TIM,DT,ENDT  STY00080
      COMMON /INT/ IGO,NBAND,NNODE,NMAT,NEL,NPEL,NGP,NELBC,NNBC,IPROB,KCSTY00090
     1,NTSC,IPUN,IPRT,ISTP                                              STY00100
      DO 110 I=1,NNODE                                                  STY00110
      PHI(I) = PHIWRK(I)                                                STY00120
      PHIWRK(I) = F(I)                                                  STY00130
  110 F(I) = -F(I)                                                      STY00140
C
```

```
C
C     FIND APPROXIMATE VALUE OF ZERO
C
      CALL BNSN (BGH,NNODE,APZRO,ABIGN)                                 STY00150
      WRITE (6,200) APZRO,ABIGN                                         STY00160
      IF (NNBC.LE.0) GO TO 150                                          STY00170
C
C     APPLY ANY KNOWN NODAL VALUES AND SOLVE EQUATION
C
      NEND = NBAND-1                                                    STY00180
      DO 140 I=1,NNBC                                                   STY00190
      ID = KNODE(I)                                                     STY00200
      NCHK = NNODE-ID                                                   STY00210
      DO 130 J=1,NEND                                                   STY00220
      IROW = ID-J                                                       STY00230
      ICOL = J+1                                                        STY00240
      INDX = ID+J                                                       STY00250
      IF (IROW.LE.0) GO TO 120                                          STY00260
      F(IROW) = F(IROW)-BGH(IROW,ICOL)*PHI(ID)                          STY00270
  120 IF (J.GT.NCHK) GO TO 130                                          STY00280
      F(INDX) = F(INDX)-BGH(ID,ICOL)*PHI(ID)                            STY00290
  130 CONTINUE                                                          STY00300
      F(ID) = F(ID)-BGH(ID,1)*PHI(ID)                                   STY00310
      BGH(ID,1) = ABIGN                                                 STY00320
  140 CONTINUE                                                          STY00330
  150 CALL LDLT (BGH,NNODE,NBAND,APZRO)                                 STY00340
      CALL SLV (BGH,F,NNODE,NBAND)                                      STY00350
      IF (NNBC.LE.0) GO TO 170                                          STY00360
      DO 160 I=1,NNBC                                                   STY00370
      ID = KNODE(I)                                                     STY00380
      PHIWRK(ID) = PHIWRK(ID)+ABIGN*F(ID)                               STY00390
  160 F(ID) = PHI(ID)                                                   STY00400
C
C     FORM TEMPERATURE VALUES AND LOAD VALUES
C
  170 WRITE (6,210)                                                     STY00410
      DO 180 I=1,NNODE                                                  STY00420
      PHI(I) = F(I)+PHIREF                                              STY00430
  180 WRITE (6,220) I,(COARD(I,J),J=1,3),PHI(I),PHIWRK(I)               STY00440
      TOT = 0.0D0                                                       STY00450
      DO 190 I=1,NNODE                                                  STY00460
  190 TOT = TOT+PHIWRK(I)                                               STY00470
      WRITE (6,230) TOT                                                 STY00480
      WHEN = 1.0D20                                                     STY00490
      IF(IPUN.NE.0) CALL PUNC(PHI,WHEN)                                 STY00491
      RETURN                                                            STY00500
  200 FORMAT ('0',10X,'THE APPROXIMATE VALUE OF ZERO = ',1PG24.16,//,23XSTY00510
     1,' ABIGN = ',G24.16,//)                                           STY00520
  210 FORMAT ('1',' NODE',7X,'X COORDINATE',8X,'Y COORDINATE',8X,'Z COORSTY00530
```

```
     1DINATE',4X,'PHI(STEADY STATE)',6X,'FORCING VECTOR',/,71X,' ADJ TO STY00540
     2REF',///)                                                           STY00550
  220 FORMAT (I5,1X,1P3G20.5,2G20.10)                                     STY00560
  230 FORMAT (///,11X,' SUM OF FORCING VECTOR ELEMENTS IS',1PG20.10,/)    STY00570
      END                                                                 STY00580
C
C*****************************************************************************
C*****************************************************************************
C
      SUBROUTINE TIMEHB                                                   TME00010
C
C     THIS SUBROUTINE SOLVES TRANSIENT HEAT EQUATION
C
      IMPLICIT REAL*8(A-H,O-Z)                                            TME00020
      COMMON /MTRX/ CORD(32,3),ENN(32,32),DCOL(32,3),COL(32),AJ(3,3),AJITME00030
     1(3,3),ENW(32,32),FE(32),PROP(278,4),COARD(278,3),F(278),PHI(278),PTME00040
     2HIDOT(278),PHIWRK(278),PHIDDT(278),BGH(278,40),BGC(278,40),PROB(27TME00050
     38,3),PUNCH(20)                                                      TME00060
      COMMON /IMTRX/ NCON(48,33),KELB(48,3),KNODE(278)                    TME00070
      COMMON /FLP/ DTJ,AKX,AKY,AKZ,FACT,PHIREF,APZRO,ABIGN,TIM,DT,ENDT   TME00080
      COMMON /INT/ IGO,NBAND,NNODE,NMAT,NEL,NPEL,NGP,NELBC,NNBC,IPROB,KCTME00090
     1,NTSC,IPUN,IPRT,ISTP                                                TME00100
      DIMENSION OUT(278,6), TIME(6), PHISAV(278,4)                        TME00110
      REWIND 10                                                           TME00120
      REWIND 11                                                           TME00130
      WRITE (10) BGC                                                      TME00140
      WRITE (11) BGH                                                      TME00150
      NLINE = 60                                                          TME00160
      IF (NLINE.GT.NNODE) NLINE=NNODE                                     TME00170
C
C     LOOP ON HEAT TRANSFER PHASES
C
      DO 350 MRH=1,NTSC                                                   TME00180
      JE = 6                                                              TME00190
      IDP = 1                                                             TME00200
      IXX = 0                                                             TME00210
      IF (MRH.EQ.1) GO TO 10                                              TME00220
      REWIND 11                                                           TME00230
      READ (11) BGH                                                       TME00240
   10 CALL INPUT (2)                                                      TME00250
      IF (NELBC.NE.0) CALL BCOND                                          TME00260
C
C     COMBINE "H" AND "C" FOR STARTING VALUES FIND APZRO, ABIGN
C
      DO 20 J=1,NBAND                                                     TME00270
      DO 20 I=1,NNODE                                                     TME00280
      CIJ = BGC(I,J)/DT                                                   TME00290
      AHIJ = BGH(I,J)/2.0D0                                               TME00300
```

```
   20 BGC(I,J) = AHIJ+CIJ                                          TME00310
      CALL BNSN (BGC,NNODE,APZRO,ABIGN)                            TME00320
      WRITE (6,360) APZRO,ABIGN                                    TME00330
      IF (NNBC.LE.0) GO TO 40                                      TME00340
C
C     APPLY ANY KNOWN NODAL TEMPERATURES
C
      DO 30 I=1,NNBC                                               TME00350
      ID = KNODE(I)                                                TME00360
      PHI(ID) = PHIWRK(ID)                                         TME00370
   30 BGC(ID,1) = ABIGN                                            TME00380
C
C     DECOMPOSE COEFFICIENT MATRIX FOR STARTING VALUES
C
   40 CALL LDLT (BGC,NNODE,NBAND,APZRO)                            TME00390
      TIME(1) = TIM                                                TME00400
      DO 50 I=1,NNODE                                              TME00410
   50 OUT(I,1) = PHI(I)+PHIREF                                     TME00420
C
C     SOLVE FOR STARTING VALUES, STORE THOSE REQUIRED
C
      DO 60 I=1,NNODE                                              TME00430
   60 PHISAV(I,1) = PHI(I)                                         TME00440
      ISTP3 = ISTP/3                                               TME00450
      ISTP2 = 2*ISTP3                                              TME00460
      DO 170 IT=1,ISTP                                             TME00470
      CALL MULT (BGH,PHI,PHIDOT,NNODE,NBAND)                       TME00480
      DO 70 I=1,NNODE                                              TME00490
   70 PHIDOT(I) = -PHIDOT(I)-F(I)                                  TME00500
      IF (NNBC.LE.0) GO TO 90                                      TME00510
      DO 80 I=1,NNBC                                               TME00520
      ID = KNODE(I)                                                TME00530
   80 PHIDOT(ID) = 0.0D0                                           TME00540
   90 CALL SLV (BGC,PHIDOT,NNODE,NBAND)                            TME00550
      DO 100 I=1,NNODE                                             TME00560
  100 PHI(I) = PHI(I)+PHIDOT(I)                                    TME00570
      IF (IT.NE.ISTP3) GO TO 130                                   TME00580
      DO 110 I=1,NNODE                                             TME00590
  110 PHISAV(I,2) = PHI(I)                                         TME00600
      IXX = IXX+1                                                  TME00610
      IF (IPRT.NE.IXX) GO TO 130                                   TME00620
      IXX = 0                                                      TME00630
      IDP = IDP+1                                                  TME00640
      TIME(IDP) = DT*IT+TIM                                        TME00650
      DO 120 I=1,NNODE                                             TME00660
  120 OUT(I,IDP) = PHI(I)+PHIREF                                   TME00670
  130 IF (IT.NE.ISTP2) GO TO 160                                   TME00680
      DO 140 I=1,NNODE                                             TME00690
```

```
  140 PHISAV(I,3) = PHI(I)                                                TME00700
      IXX = IXX+1                                                         TME00710
      IF (IPRT.NE.IXX) GO TO 160                                          TME00720
      IXX = 0                                                             TME00730
      IDP = IDP+1                                                         TME00740
      TIME(IDP) = DT*IT+TIM                                               TME00750
      DO 150 I=1,NNODE                                                    TME00760
  150 OUT(I,IDP) = PHI(I)+PHIREF                                          TME00770
  160 WHEN = TIM+DT*IT                                                    TME00780
      IF(IPUN.NE.0) CALL PUNC(PHI,WHEN)                                   TME00781
  170 CONTINUE                                                            TME00790
      DO 180 I=1,NNODE                                                    TME00800
  180 PHISAV(I,4) = PHI(I)                                                TME00810
      IXX = IXX+1                                                         TME00820
      IF (IPRT.NE.IXX) GO TO 200                                          TME00830
      IXX = 0                                                             TME00840
      IDP = IDP+1                                                         TME00850
      TIME(IDP) = DT*IT+TIM                                               TME00860
      DO 190 I=1,NNODE                                                    TME00870
  190 OUT(I,IDP) = PHI(I)+PHIREF                                          TME00880
C
C     REBUILD LEFT HAND SIDE FOR FIVE POINT FINITE DIFFERENCE METHOD
C     FIND ABIGN,APZRO
C
  200 REWIND 10                                                           TME00890
      READ (10) BGC                                                       TME00900
      DT = DT*DFLOAT(ISTP3)                                               TME00910
      ITIM = (ENDT-TIM)/DT                                                TME00920
      NTP = (ITIM/IPRT)+1                                                 TME00930
      DO 210 J=1,NBAND                                                    TME00940
      DO 210 I=1,NNODE                                                    TME00950
      CIJ = BGC(I,J)*25.0D0/DT                                            TME00960
      AHIJ = BGH(I,J)*12.0D0                                              TME00970
  210 BGH(I,J) = CIJ+AHIJ                                                 TME00980
      CALL BNSN (BGH,NNODE,APZRO,ABIGN)                                   TME00990
      IF (NNBC.EQ.0) GO TO 230                                            TME01000
C
C     APPLY ANY KNOWN NODAL VALUES
C
      DO 220 I=1,NNBC                                                     TME01010
      ID = KNODE(I)                                                       TME01020
  220 BGH(ID,1) = ABIGN                                                   TME01030
  230 CALL LDLT (BGH,NNODE,NBAND,APZRO)                                   TME01040
C
C     COMMENCE SOLUTION, PRINTING AND PUNCHING OUTPUT AS REQUIRED
C
      DO 340 IT=4,ITIM                                                    TME01050
      DO 240 I=1,NNODE                                                    TME01060
```

```
240 PHI(I) = (48.0D0*PHISAV(I,4)-36.0D0*PHISAV(I,3)+16.0D0*PHISAV(I,2)
   1-3.0D0*PHISAV(I,1))/DT
    CALL MULT (BGC,PHI,PHIDOT,NNODE,NBAND)
    DO 250 I=1,NNODE
250 PHI(I) = PHIDOT(I)-12.0D0*F(I)
    IF (NNBC.EQ.0) GO TO 270
    DO 260 I=1,NNBC
    ID = KNODE(I)
260 PHI(ID) = ABIGN*PHIWRK(ID)
270 CALL SLV (BGH,PHI,NNODE,NBAND)
    DO 280 I=1,NNODE
    PHISAV(I,1) = PHISAV(I,2)
    PHISAV(I,2) = PHISAV(I,3)
    PHISAV(I,3) = PHISAV(I,4)
280 PHISAV(I,4) = PHI(I)
    IXX = IXX+1
    IF (IXX.NE.IPRT) GO TO 330
    IXX = 0
    IDP = IDP+1
    TIME(IDP) = DT*IT+TIM
    DO 290 I=1,NNODE
290 OUT(I,IDP) = PHI(I)+PHIREF
    IF (NTP.LT.JE) JE = NTP
    IF (IDP.LT.JE) GO TO 330
    NL1 = 1
    NL2 = NLINE
    WRITE (6,370)
300 WRITE (6,380) DT,(TIME(J),J=1,JE)
    WRITE (6,390)
    DO 310 I=NL1,NL2
310 WRITE (6,400) I,(COARD(I,J),J=1,3),(OUT(I,K),K=1,JE)
    IF (NL2.EQ.NNODE) GO TO 320
    NL1 = NL2+1
    NL2 = NL2+NLINE
    IF (NL2.GT.NNODE) NL2=NNODE
    WRITE (6,410)
    GO TO 300
320 NTP = NTP-JE
    IDP = 0
330 WHEN = TIM+DT*IT
    IF(IPUN.NE.0) CALL PUNC(PHI,WHEN)
340 CONTINUE
350 CONTINUE
    RETURN
360 FORMAT (///,11X,'THE APPROXIMATE VALUE(FOR STARTING VALUES) OF ZER
   1O = ',1PG24.16,//,23X,' ABIGN =',G24.16,//)
370 FORMAT ('1',38X,'DISTRIBUTION OF TEMPERATURE VERSUS TIME/SPATIAL P
   1OSITION',/,42X,'FINITE DIFFERENCE METHOD WITH TRAPEZOIDAL STARTER'
```

```
     2//)                                                                TME01550
 380 FORMAT (/,70X,'TIME OF PHI DISTRIBUTION  ',//,70X,'DELTA TIME =',FTME01560
    110.5,//,41X,1P6G15.6)                                               TME01570
 390 FORMAT (//,' NODE',2X,'X CORD',6X,'Y CORD',6X,'Z CORD',2X,6(9X,'PHTME01580
    1I',3X),//)                                                          TME01590
 400 FORMAT (I5,1P3G12.4,6G15.7)                                         TME01600
 410 FORMAT ('1')                                                        TME01610
     END                                                                 TME01620
C
C***********************************************************************
C***********************************************************************
C
      SUBROUTINE BCOND                                                   BCD00010
C
C     THIS SUBROUTINE APPLIES ALL BOUNDARY CONDITIONS AND INTERNAL
C     GENERATION
C
      IMPLICIT REAL*8(A-H,O-Z)                                           BCD00020
      COMMON /MTRX/ CORD(32,3),ENN(32,32),DCOL(32,3),COL(32),AJ(3,3),AJIBCD00030
     1(3,3),ENW(32,32),FE(32),PROP(278,4),COARD(278,3),F(278),PHI(278),PBCD00040
     2HIDOT(278),PHIWRK(278),PHIDDT(278),BGH(278,40),BGC(278,40),PROB(27BCD00050
     38,3),PUNCH(20)                                                     BCD00060
      COMMON /IMTRX/ NCON(48,33),KELB(48,3),KNODE(278)                   BCD00070
      COMMON /FLP/ DTJ,AKX,AKY,AKZ,FACT,PHIREF,APZRO,ABIGN,TIM,DT,ENDT  BCD00080
      COMMON /INT/ IGO,NBAND,NNODE,NMAT,NEL,NPEL,NGP,NELBC,NNBC,IPROB,KCBCD00090
     1,NTSC,IPUN,IPRT,ISTP                                               BCD00100
      DO 10 I=1,NNODE                                                    BCD00110
   10 F(I) = 0.0D0                                                       BCD00120
      DO 90 I=1,NELBC                                                    BCD00130
      DO 30 II=1,NPEL                                                    BCD00140
      FE(II) = 0.0D0                                                     BCD00150
      DO 20 JJ=1,3                                                       BCD00160
   20 CORD(II,JJ) = 0.0D0                                                BCD00170
      DO 30 KK=1,NPEL                                                    BCD00180
   30 ENW(II,KK) = 0.0D0                                                 BCD00190
      IEL = KELB(I,1)                                                    BCD00200
      KBT = KELB(I,2)                                                    BCD00210
      KBS = KELB(I,3)                                                    BCD00220
      DO 40 J=1,NPEL                                                     BCD00230
      INDX = NCON(IEL,J+1)                                               BCD00240
      DO 40 K=1,3                                                        BCD00250
   40 CORD(J,K) = COARD(INDX,K)                                          BCD00260
      CALL CUBEB (NPEL,NGP,KBT,KBS,IEL)                                  BCD00270
      GO TO (50,50,70), KBT                                              BCD00280
C
C     ASSEMBLE TYPE 1 OR 2 B.C.
C
   50 DO 60 L=1,NPEL                                                     BCD00290
```

```
      ID = NCON(IEL,L+1)                                                BCD00300
   60 F(ID) = F(ID)+FE(L)                                               BCD00310
      GO TO 90                                                          BCD00320
C
C     ASSEMBLE TYPE 3 B.C.
C
   70 DO 80 KK=1,NPEL                                                   BCD00330
      IDX = NCON(IEL,KK+1)                                              BCD00340
      DO 80 LL=1,NPEL                                                   BCD00350
      IDY = NCON(IEL,LL+1)-IDX+1                                        BCD00360
      IF (IDY.LT.1) GO TO 80                                            BCD00370
      BGH(IDX,IDY) = BGH(IDX,IDY)+ENW(KK,LL)                            BCD00380
   80 CONTINUE                                                          BCD00390
   90 CONTINUE                                                          BCD00400
      RETURN                                                            BCD00410
      END                                                               BCD00420
C
C*************************************************************************
C*************************************************************************
C
      SUBROUTINE CUBE (NPEL,NGP,ITYPE,IEL,NMAT)                         CUB00010
C
C     THIS SUBROUTINE PERFORMS THREE DIMENSIONAL INTEGRATION FOR "H"
C     AND "C" MATRICES
C
      IMPLICIT REAL*8(A-H,O-Z)                                          CUB00020
      COMMON /MTRX/ CORD(32,3),ENN(32,32),DCOL(32,3),COL(32),AJ(3,3),AJICUB00030
     1(3,3),ENW(32,32),FE(32),PROP(278,4),CUARD(278,3),F(278),PHI(278),PCUB00040
     2HIDOT(278),PHIWRK(278),PHIDDT(278),BGH(278,40),BGC(278,40),PROB(27CUB00050
     38,3),PUNCH(20)                                                    CUB00060
      COMMON /IMTRX/ NCON(48,33),KELB(48,3),KNODE(278)                  CUB00070
      COMMON /FLP/ DTJ,AKX,AKY,AKZ,FACT,PHIREF,APZRO,ABIGN,TIM,DT,ENDT  CUB00080
      DIMENSION GP(6,5), WF(6,5)                                        CUB00090
      DATA GP/.577350269189630D0,-.577350269189630D0,0.0D0,0.0D0,0.0D0,0.DCUB00100
     10,.774596669241480D0,0.0D0,-.774596669241480D0,0.0D0,0.0D0,0.0D0,.86CUB00110
     2113631159405D0,0.339981043584860D0,-.339981043584860D0,-.861136311594CUB00120
     3405D0,0.0D0,0.0D0,.906179845938660D0,.538469310105680D0,0.0D0,-.5384CUB00130
     4693101056800,-.906179845938660D0,0.0D0,.932469514203150D0,.661209386CUB00140
     5466260D0,.238619186083170D0,-.238619186083170D0,-.661209386466260D0,-.CUB00150
     6932469514203150D0/                                                CUB00160
      DATA WF/1.0D0,1.0D0,0.0D0,0.0D0,0.0D0,0.0D0,.555555555555560D0,.888CUB00170
     1888888888890D0,.555555555555560D0,0.0D0,0.0D0,0.0D0,.347854845137450CUB00180
     20,.652145154862550D0,.652145154862550D0,.347854845137450D0,0.0D0,0.0DCUB00190
     30,.236926885056190D0,.478628670499370D0,.568888888888890D0,.4786286700CUB00200
     4499370D0,.236926885056190D0,0.0D0,.171324492379170D0,.360761573048140CUB00210
     50,.467913934572690D0,.467913934572690D0,.360761573048140D0,.171324492CUB00220
     6379170D0/                                                         CUB00230
      IF (NGP.LE.0.OR.NGP.GT.6) NGP=3                                   CUB00240
```

```
      IDGP = NGP-1                                             CUB00250
      DO 40 I=1,NGP                                            CUB00260
      XI = GP(I,IDGP)                                          CUB00270
      WX = WF(I,IDGP)                                          CUB00280
      DO 40 J=1,NGP                                            CUB00290
      ETA = GP(J,IDGP)                                         CUB00300
      WY = WF(J,IDGP)                                          CUB00310
      DO 40 K=1,NGP                                            CUB00320
      ZETA = GP(K,IDGP)                                        CUB00330
      WZ = WF(K,IDGP)                                          CUB00340
      WFACT = WX*WY*WZ                                         CUB00350
      IF (ITYPE.EQ.1) GO TO 10                                 CUB00360
      CALL FORMC (XI,ETA,ZETA,NPEL,IEL,NMAT)                   CUB00370
      GO TO 20                                                 CUB00380
   10 CALL FORMH (XI,ETA,ZETA,NPEL,IEL,NMAT)                   CUB00390
   20 DO 30 L=1,NPEL                                           CUB00400
      DO 30 M=1,NPEL                                           CUB00410
   30 ENW(L,M) = ENW(L,M)+WFACT*ENN(L,M)                       CUB00420
   40 CONTINUE                                                 CUB00430
      DO 50 I=1,NPEL                                           CUB00440
      DO 50 J=I,NPEL                                           CUB00450
      ENW(I,J) = 5.D-1*(ENW(I,J)+ENW(J,I))                     CUB00460
   50 ENW(J,I) = ENW(I,J)                                      CUB00470
      RETURN                                                   CUB00480
C
C     FOR USE WITH BOUNDARY CONDITIONS
C
      ENTRY CUBEB(NPEL,NGP,KBT,KBS,IEL)                        CUB00490
      NGPT = NGP                                               CUB00500
      NGP = 5                                                  CUB00510
      IDGP = NGP-1                                             CUB00520
      KBSABS = IABS(KBS)                                       CUB00530
      GO TO (60,100,100), KBT                                  CUB00540
   60 DO 90 I=1,NGP                                            CUB00550
      XI = GP(I,IDGP)                                          CUB00560
      WX = WF(I,IDGP)                                          CUB00570
      DO 90 J=1,NGP                                            CUB00580
      ETA = GP(J,IDGP)                                         CUB00590
      WY = WF(J,IDGP)                                          CUB00600
      DO 90 K=1,NGP                                            CUB00610
      ZETA = GP(K,IDGP)                                        CUB00620
      WZ = WF(K,IDGP)                                          CUB00630
      WFACT = WX*WY*WZ                                         CUB00640
      CALL FORMV (XI,ETA,ZETA,NPEL)                            CUB00650
      CONST = 0.0D0                                            CUB00660
      DO 70 L=1,NPEL                                           CUB00670
      ID = NCON(IEL,L+1)                                       CUB00680
   70 CONST = CONST+PROB(ID,1)*COL(L)                          CUB00690
```

```
      CONST = CONST*DTJ*WFACT
      DO 80 L=1,NPEL
   80 FE(L) = FE(L)-COL(L)*CONST
   90 CONTINUE
      GO TO 220
  100 C3 = DFLOAT(KBS/KBSABS)
      DO 200 I=1,NGP
      C1 = GP(I,IDGP)
      W1 = WF(I,IDGP)
      DO 200 J=1,NGP
      C2 = GP(J,IDGP)
      W2 = WF(J,IDGP)
      WFACT = W1*W2
      GO TO (110,120,130), KBSABS
  110 XI = C3
      ETA = C1
      ZETA = C2
      GO TO 140
  120 XI = C1
      ETA = C3
      ZETA = C2
      GO TO 140
  130 XI = C1
      ETA = C2
      ZETA = C3
  140 IF (KBT.EQ.3) GO TO 170
      CALL FORMSQ (XI,ETA,ZETA,NPEL,KBSABS)
      CONST = 0.0D0
      DO 150 L=1,NPEL
      ID = NCON(IEL,L+1)
  150 CONST = CONST+PROB(ID,2)*COL(L)
      CONST = CONST*DTJ*WFACT
      DO 160 L=1,NPEL
  160 FE(L) = FE(L)+CONST*COL(L)
      GO TO 200
  170 CALL FORMSA (XI,ETA,ZETA,NPEL,KBSABS)
      CONST = 0.0D0
      DO 180 L=1,NPEL
      ID = NCON(IEL,L+1)
  180 CONST = CONST+PROB(ID,3)*COL(L)
      CONST = CONST*WFACT*DTJ
      DO 190 L=1,NPEL
      DO 190 M=1,NPEL
  190 ENW(L,M) = ENW(L,M)+ENN(L,M)*CONST
  200 CONTINUE
      IF (KBT.NE.3) GO TO 220
      DO 210 I=1,NPEL
      DO 210 J=I,NPEL
```

```
      ENW(I,J) = 5.D-1*(ENW(I,J)+ENW(J,I))                              CUB01180
  210 ENW(J,I) = ENW(I,J)                                               CUB01190
  220 NGP = NGPT                                                        CUB01200
      RETURN                                                            CUB01210
      END                                                               CUB01220
C
C***********************************************************************
C***********************************************************************
C
      SUBROUTINE FORMC (XI,ETA,ZETA,NPEL,IEL,NMAT)                      FMC00010
C
C     THIS SUBROUTINE BUILDS ELEMENTAL "H" AND "C" MATRICES
C
      IMPLICIT REAL*8(A-H,O-Z)                                          FMC00020
      COMMON /MTRX/ CORD(32,3),ENN(32,32),DCOL(32,3),COL(32),AJ(3,3),AJIFMC00030
     1(3,3),ENW(32,32),FE(32),PROP(278,4),COARD(278,3),F(278),PHI(278),PFMC00040
     2HIDOT(278),PHIWRK(278),PHIDDT(278),BGH(278,40),BGC(278,40),PROB(27FMC00050
     38,3),PUNCH(20)                                                    FMC00060
      COMMON /IMTRX/ NCON(48,33),KELB(48,3),KNODE(278)                  FMC00070
      COMMON /FLP/ DTJ,AKX,AKY,AKZ,FACT,PHIREF,APZRO,ABIGN,TIM,DT,ENDT  FMC00080
C
C     FORM "C"
C
      CALL SHAPE (XI,ETA,ZETA,NPEL)                                     FMC00090
      CALL JACOB (NPEL,AJ,DCOL,CORD,AJI,DTJ)                            FMC00100
      RHOC = FACT                                                       FMC00110
      IF (NMAT.GT.0) GO TO 120                                          FMC00120
      RHOC = 0.0D0                                                      FMC00130
      DO 110 I=1,NPEL                                                   FMC00140
      ID = NCON(IEL,I+1)                                                FMC00150
  110 RHOC = RHOC+PROP(ID,4)*COL(I)                                     FMC00160
  120 DO 130 I=1,NPEL                                                   FMC00170
      A = COL(I)*DTJ                                                    FMC00180
      DO 130 J=1,NPEL                                                   FMC00190
  130 ENN(I,J) = A*COL(J)*RHOC                                          FMC00200
      RETURN                                                            FMC00210
C
C     FORM "H"
C
      ENTRY FORMH(XI,ETA,ZETA,NPEL,IEL,NMAT)                            FMC00220
      CALL SHAPE (XI,ETA,ZETA,NPEL)                                     FMC00230
      CALL JACOB (NPEL,AJ,DCOL,CORD,AJI,DTJ)                            FMC00240
      DO 140 I=1,NPEL                                                   FMC00250
      DNX = DCOL(I,1)                                                   FMC00260
      DNY = DCOL(I,2)                                                   FMC00270
      DNZ = DCOL(I,3)                                                   FMC00280
      DCOL(I,1) = AJI(1,1)*DNX+AJI(1,2)*DNY+AJI(1,3)*DNZ                FMC00290
      DCOL(I,2) = AJI(2,1)*DNX+AJI(2,2)*DNY+AJI(2,3)*DNZ                FMC00300
```

```
  140 DCOL(I,3) = AJI(3,1)*DNX+AJI(3,2)*DNY+AJI(3,3)*DNZ              FMC00310
      IF (NMAT.GT.0) GO TO 160                                          FMC00320
      AKX = 0.0D0                                                       FMC00330
      DO 150 I=1,NPEL                                                   FMC00340
      ID = NCON(IEL,I+1)                                                FMC00350
  150 AKX = AKX+PROP(ID,1)*COL(I)                                       FMC00360
  160 DO 170 I=1,NPEL                                                   FMC00370
      BX = DCOL(I,1)*DTJ                                                FMC00380
      DO 170 J=1,NPEL                                                   FMC00390
  170 ENN(I,J) = BX*AKX*DCOL(J,1)                                       FMC00400
      IF (NMAT.GT.0) GO TO 190                                          FMC00410
      AKY = 0.0D0                                                       FMC00420
      DO 180 I=1,NPEL                                                   FMC00430
      ID = NCON(IEL,I+1)                                                FMC00440
  180 AKY = AKY+PROP(ID,2)*COL(I)                                       FMC00450
  190 DO 200 I=1,NPEL                                                   FMC00460
      BY = DCOL(I,2)*DTJ                                                FMC00470
      DO 200 J=1,NPEL                                                   FMC00480
  200 ENN(I,J) = BY*AKY*DCOL(J,2)+ENN(I,J)                              FMC00490
      IF (NMAT.GT.0) GO TO 220                                          FMC00500
      AKZ = 0.0D0                                                       FMC00510
      DO 210 I=1,NPEL                                                   FMC00520
      ID = NCON(IEL,I+1)                                                FMC00530
  210 AKZ = AKZ+PROP(ID,3)*COL(I)                                       FMC00540
  220 DO 230 I=1,NPEL                                                   FMC00550
      BZ = DCOL(I,3)*DTJ                                                FMC00560
      DO 230 J=1,NPEL                                                   FMC00570
  230 ENN(I,J) = BZ*AKZ*DCOL(J,3)+ENN(I,J)                              FMC00580
      RETURN                                                            FMC00590
C
C     FOR USE WITH B.C. CALCULATIONS
C
      ENTRY FORMV(XI,ETA,ZETA,NPEL)                                     FMC00600
      CALL SHAPE (XI,ETA,ZETA,NPEL)                                     FMC00610
      CALL JACOB (NPEL,AJ,DCOL,CORD,AJI,DTJ)                            FMC00620
      RETURN                                                            FMC00630
      ENTRY FORMSQ(XI,ETA,ZETA,NPEL,KBSABS)                             FMC00640
      CALL SHAPE (XI,ETA,ZETA,NPEL)                                     FMC00650
      CALL JACOBB (NPEL,AJ,DCOL,CORD,DTJ,KBSABS)                        FMC00660
      RETURN                                                            FMC00670
      ENTRY FORMSA(XI,ETA,ZETA,NPEL,KBSABS)                             FMC00680
      CALL SHAPE (XI,ETA,ZETA,NPEL)                                     FMC00690
      CALL JACOBB (NPEL,AJ,DCOL,CORD,DTJ,KBSABS)                        FMC00700
      DO 240 I=1,NPEL                                                   FMC00710
      A = COL(I)                                                        FMC00720
      DO 240 J=1,NPEL                                                   FMC00730
  240 ENN(I,J) = A*COL(J)                                               FMC00740
      RETURN                                                            FMC00750
```

```
      END                                                               FMC00760
C
C***********************************************************************
C***********************************************************************
C
      SUBROUTINE SHAPE (XI,ETA,ZETA,NPEL)                               SHP00010
C
C     THIS SUBROUTINE CALCULATES SHAPE FUNCTION VALUES
C
      IMPLICIT REAL*8(A-H,O-Z)                                          SHP00020
      COMMON /MTRX/ CORD(32,3),ENN(32,32),DCOL(32,3),COL(32),AJ(3,3),AJISHP00030
     1(3,3),ENW(32,32),FE(32),PROP(278,4),COARD(278,3),F(278),PHI(278),PSHP00040
     2HIDOT(278),PHIWRK(278),PHIDDT(278),BGH(278,40),BGC(278,40),PROB(27SHP00050
     38,3),PUNCH(20)                                                    SHP00060
      COMMON /IMTRX/ NCON(48,33),KELB(48,3),KNODE(278)                  SHP00070
      COMMON /FLP/ DTJ,AKX,AKY,AKZ,FACT,PHIREF,APZRO,ABIGN,TIM,DT,ENDT  SHP00080
      DIMENSION CORDX(4), CORDY(4), CORDZ(4), TRD(2)                    SHP00090
      DATA CORDX/1.0D0,-1.0D0,-1.0D0,1.0D0/                             SHP00100
      DATA CORDY/1.0D0,1.0D0,-1.0D0,-1.0D0/                             SHP00110
      DATA CORDZ/1.0D0,-1.0D0,1.0D0,-1.0D0/                             SHP00120
      DATA TRD/.33333333333333333D0,-.33333333333333333D0/              SHP00130
      FL(XI,ETA,ZETA,X1,Y1,Z1)=(1.0D0+XI*X1)*(1.0D0+ETA*Y1)*(1.0D0+ZETA*SHP00140
     1Z1)/8.0D0                                                         SHP00150
      DFL(ETA,ZETA,X1,Y1,Z1)=X1*(1.0D0+ETA*Y1)*(1.0D0+ZETA*Z1)/8.0D0    SHP00160
      FQC(XI,ETA,ZETA,X1,Y1,Z1)=(1.0D0+XI*X1)*(1.0D0+ETA*Y1)*(1.0D0+ZETASHP00170
     1*Z1)*(XI*X1+ETA*Y1+ZETA*Z1-2.0D0)/8.0D0                           SHP00180
      FQM(XI,ETA,ZETA,Y1,Z1)=(1.0D0-XI*XI)*(1.0D0+ETA*Y1)*(1.0D0+ZETA*Z1SHP00190
     1)/4.0D0                                                           SHP00200
      DFQC(XI,ETA,ZETA,X1,Y1,Z1)=X1*(1.0D0+ETA*Y1)*(1.0D0+ZETA*Z1)*(2.0DSHP00210
     10*XI*X1+ETA*Y1+ZETA*Z1-1.0D0)/8.0D0                               SHP00220
      DFQMX(XI,ETA,ZETA,Y1,Z1)=-XI*(1.0D0+ETA*Y1)*(1.0D0+ZETA*Z1)/2.0D0 SHP00230
      DFQMY(XI,ZETA,Y1,Z1)=Y1*(1.0D0-XI*XI)*(1.0D0+ZETA*Z1)/4.0D0       SHP00240
      FCC(XI,ETA,ZETA,X1,Y1,Z1)=(1.0D0+XI*X1)*(1.0D0+ETA*Y1)*(1.0D0+ZETASHP00250
     1*Z1)*(9.0D0*(XI*XI+ETA*ETA+ZETA*ZETA)-1.9D1)/6.4D1                SHP00260
      FCM(XI,ETA,ZETA,X1,Y1,Z1)=9.0D0*(1.0D0-XI*XI)*(1.0D0+9.0D0*XI*X1)*SHP00270
     1(1.0D0+ETA*Y1)*(1.0D0+ZETA*Z1)/6.4D1                              SHP00280
      DFCC(XI,ETA,ZETA,X1,Y1,Z1)=(1.0D0+ETA*Y1)*(1.0D0+ZETA*Z1)*(9.0D0*XSHP00290
     11*(XI*XI+ETA*ETA+ZETA*ZETA)-1.9D1*X1+1.8D1*XI+1.8D1*XI*XI*X1)/6.4DSHP00300
     21                                                                 SHP00310
      DFCMX(XI,ETA,ZETA,X1,Y1,Z1)=9.0D0*(1.0D0+ETA*Y1)*(1.0D0+ZETA*Z1)*(SHP00320
     1-2.0D0*XI-2.7D1*XI*XI*X1+9.0D0*X1)/6.4D1                          SHP00330
      DFCMY(XI,ZETA,X1,Y1,Z1)=9.0D0*(1.0D0-XI*XI)*(1.0D0+9.0D0*XI*X1)*Y1SHP00340
     1*(1.0D0+ZETA*Z1)/6.4D1                                            SHP00350
      IGO = (NPEL-8)/12+1                                               SHP00360
      GO TO (110,130,160), IGO                                          SHP00370
C
C     LINEAR FUNCTIONS
C
```

```
  110 ICOL = 0                                                     SHP00380
      DO 120 IZ=1,2                                                SHP00390
      DO 120 IX=1,4                                                SHP00400
      ICOL = ICOL+1                                                SHP00410
      X1 = CORDX(IX)                                               SHP00420
      Y1 = CORDY(IX)                                               SHP00430
      Z1 = CORDZ(IZ)                                               SHP00440
      COL(ICOL) = FL(XI,ETA,ZETA,X1,Y1,Z1)                         SHP00450
      DCOL(ICOL,1) = DFL(ETA,ZETA,XI,Y1,Z1)                        SHP00460
      DCOL(ICOL,2) = DFL(ZETA,XI,Y1,Z1,X1)                         SHP00470
  120 DCOL(ICOL,3) = DFL(XI,ETA,Z1,X1,Y1)                          SHP00480
      GO TO 190                                                    SHP00490
C
C     QUADRATIC FUNCTIONS
C     CORNER NODES
C
  130 DO 140 IZ=1,2                                                SHP00500
      II = 12*(IZ-1)+1                                             SHP00510
      IT = II+6                                                    SHP00520
      IX = 0                                                       SHP00530
      DO 140 ICOL=II,IT,2                                          SHP00540
      IX = IX+1                                                    SHP00550
      Y1 = CORDY(IX)                                               SHP00560
      X1 = CORDX(IX)                                               SHP00570
      Z1 = CORDZ(IZ)                                               SHP00580
      COL(ICOL) = FQC(XI,ETA,ZETA,X1,Y1,Z1)                        SHP00590
      DCOL(ICOL,1) = DFQC(XI,ETA,ZETA,X1,Y1,Z1)                    SHP00600
      DCOL(ICOL,2) = DFQC(ETA,ZETA,XI,Y1,Z1,X1)                    SHP00610
  140 DCOL(ICOL,3) = DFQC(ZETA,XI,ETA,Z1,X1,Y1)                    SHP00620
      IC9 = 8                                                      SHP00630
C
C     MID-SIDE NODES
C
      DO 150 IZ=1,2                                                SHP00640
      II = 12*(IZ-1)+2                                             SHP00650
      IT = II+4                                                    SHP00660
      DO 150 IC2=II,IT,4                                           SHP00670
      IC4 = IC2+2                                                  SHP00680
      IC9 = IC9+1                                                  SHP00690
      IN = IC9-8                                                   SHP00700
      X1 = CORDX(IN)                                               SHP00710
      Y1 = CORDY(IN)                                               SHP00720
      Z1 = CORDZ(IN)                                               SHP00730
      COL(IC2) = FQM(XI,ETA,ZETA,Z1,Y1)                            SHP00740
      COL(IC4) = FQM(ETA,ZETA,XI,Y1,-Z1)                           SHP00750
      COL(IC9) = FQM(ZETA,XI,ETA,X1,Y1)                            SHP00760
      DCOL(IC2,1) = DFQMX(XI,ETA,ZETA,Z1,Y1)                       SHP00770
      DCOL(IC2,2) = DFQMY(XI,ZETA,Z1,Y1)                           SHP00780
```

```
      DCOL(IC2,3) = DFQMY(XI,ETA,Y1,Z1)                                 SHP00790
      DCOL(IC4,1) = DFQMY(ETA,ZETA,-Z1,Y1)                              SHP00800
      DCOL(IC4,2) = DFQMX(ETA,XI,ZETA,-Z1,Y1)                           SHP00810
      DCOL(IC4,3) = DFQMY(ETA,XI,Y1,-Z1)                                SHP00820
      DCOL(IC9,1) = DFQMY(ZETA,ETA,X1,Y1)                               SHP00830
      DCOL(IC9,2) = DFQMY(ZETA,XI,Y1,X1)                                SHP00840
  150 DCOL(IC9,3) = DFQMX(ZETA,XI,ETA,X1,Y1)                            SHP00850
      GO TO 190                                                         SHP00860
C
C     CUBIC FUNCTIONS
C     CORNER NODES
C
  160 DO 170 IZ=1,2                                                     SHP00870
      II = 20*(IZ-1)+1                                                  SHP00880
      IT = II+9                                                         SHP00890
      IX = 0                                                            SHP00900
      DO 170 ICOL=II,IT,3                                               SHP00910
      IX = IX+1                                                         SHP00920
      X1 = CORDX(IX)                                                    SHP00930
      Y1 = CORDY(IX)                                                    SHP00940
      Z1 = CORDZ(IZ)                                                    SHP00950
      COL(ICOL) = FCC(XI,ETA,ZETA,X1,Y1,Z1)                             SHP00960
      DCOL(ICOL,1) = DFCC(XI,ETA,ZETA,X1,Y1,Z1)                         SHP00970
      DCOL(ICOL,2) = DFCC(ETA,ZETA,XI,Y1,Z1,X1)                         SHP00980
  170 DCOL(ICOL,3) = DFCC(ZETA,XI,ETA,Z1,X1,Y1)                         SHP00990
      I13 = 12                                                          SHP01000
C
C     MID-SIDE NODES
C
      DO 180 IZ=1,2                                                     SHP01010
      ID = 9-2*IZ                                                       SHP01020
      IX = 0                                                            SHP01030
      DO 180 II=1,2                                                     SHP01040
      IS = 20*II-19+IZ                                                  SHP01050
      IT = 20*II-10-IZ                                                  SHP01060
      DO 180 ICOL=IS,IT,ID                                              SHP01070
      I13 = I13+1                                                       SHP01080
      I5 = ICOL+3                                                       SHP01090
      IX = IX+1                                                         SHP01100
      X1 = CORDX(IX)                                                    SHP01110
      Y1 = CORDY(IX)                                                    SHP01120
      Z1 = CORDZ(IX)                                                    SHP01130
      THIRD = TRD(IZ)                                                   SHP01140
      COL(ICOL) = FCM(XI,ETA,ZETA,THIRD,Z1,Y1)                          SHP01150
      COL(I5) = FCM(ETA,ZETA,XI,THIRD,Y1,-Z1)                           SHP01160
      COL(I13) = FCM(ZETA,XI,ETA,THIRD,X1,Y1)                           SHP01170
      DCOL(ICOL,1) = DFCMX(XI,ETA,ZETA,THIRD,Z1,Y1)                     SHP01180
      DCOL(ICOL,2) = DFCMY(XI,ZETA,THIRD,Z1,Y1)                         SHP01190
```

```
      DCOL(ICOL,3) = DFCMY(XI,ETA,THIRD,Y1,Z1)                          SHP01200
      DCOL(I5,1) = DFCMY(ETA,ZETA,THIRD,-Z1,Y1)                         SHP01210
      DCOL(I5,2) = DFCMX(ETA,ZETA,XI,THIRD,Y1,-Z1)                      SHP01220
      DCOL(I5,3) = DFCMY(ETA,XI,THIRD,Y1,-Z1)                           SHP01230
      DCOL(I13,1) = DFCMY(ZETA,ETA,THIRD,X1,Y1)                         SHP01240
      DCOL(I13,2) = DFCMY(ZETA,XI,THIRD,Y1,X1)                          SHP01250
  180 DCOL(I13,3) = DFCMX(ZETA,XI,ETA,THIRD,X1,Y1)                      SHP01260
  190 RETURN                                                            SHP01270
      END                                                               SHP01280
C
C***********************************************************************
C***********************************************************************
C
      SUBROUTINE JACOB (NPEL,AJ,DCOL,CORD,AJI,DTJ)                      JCB00010
C
C     THIS SUBROUTINE WILL FORM THE JACOBIAN AND ITS DETERMINANT
C
      IMPLICIT REAL*8(A-H,O-Z)                                          JCB00020
      DIMENSION AJ(3,3), AJI(3,3), CORD(32,3), DCOL(32,3)               JCB00030
C
C     FORM JACOBIAN
C
      DO 110 I=1,3                                                      JCB00040
      DO 110 J=1,3                                                      JCB00050
      AJ(I,J) = 0.0D0                                                   JCB00060
      DO 110 K=1,NPEL                                                   JCB00070
  110 AJ(I,J) = AJ(I,J)+DCOL(K,I)*CORD(K,J)                             JCB00080
C
C     COMPUTE DETERMINANT
C
      DTJ = AJ(1,1)*AJ(2,2)*AJ(3,3)+AJ(1,2)*AJ(2,3)*AJ(3,1)+AJ(1,3)*AJ(2JCB00090
     1,1)*AJ(3,2)-AJ(3,1)*AJ(2,2)*AJ(1,3)-AJ(3,2)*AJ(2,3)*AJ(1,1)-AJ(3,3JCB00100
     2)*AJ(2,1)*AJ(1,2)                                                 JCB00110
      AJI(1,1) = (AJ(2,2)*AJ(3,3)-AJ(3,2)*AJ(2,3))/DTJ                  JCB00120
      AJI(2,1) = -(AJ(2,1)*AJ(3,3)-AJ(3,1)*AJ(2,3))/DTJ                 JCB00130
      AJI(3,1) = (AJ(2,1)*AJ(3,2)-AJ(3,1)*AJ(2,2))/DTJ                  JCB00140
      AJI(1,2) = -(AJ(1,2)*AJ(3,3)-AJ(3,2)*AJ(1,3))/DTJ                 JCB00150
      AJI(2,2) = (AJ(1,1)*AJ(3,3)-AJ(3,1)*AJ(1,3))/DTJ                  JCB00160
      AJI(3,2) = -(AJ(1,1)*AJ(3,2)-AJ(3,1)*AJ(1,2))/DTJ                 JCB00170
      AJI(1,3) = (AJ(1,2)*AJ(2,3)-AJ(2,2)*AJ(1,3))/DTJ                  JCB00180
      AJI(2,3) = -(AJ(1,1)*AJ(2,3)-AJ(2,1)*AJ(1,3))/DTJ                 JCB00190
      AJI(3,3) = (AJ(1,1)*AJ(2,2)-AJ(2,1)*AJ(1,2))/DTJ                  JCB00200
      RETURN                                                            JCB00210
C
C     FOR USE WITH B.C. CALCULATIONS
C
      ENTRY JACOBB(NPEL,AJ,DCOL,CORD,DTJ,KS)                            JCB00220
      DO 120 I=1,3                                                      JCB00230
```

```
      DO 120 J=1,3                                                      JCB00240
      AJ(I,J) = 0.0D0                                                   JCB00250
      DO 120 K=1,NPEL                                                   JCB00260
  120 AJ(I,J) = AJ(I,J)+DCOL(K,I)*CORD(K,J)                             JCB00270
C
C     CCMPUTE SURFACE INTEGRAL TRANSFORMATION MULTIPLIER
C
      GO TO (130,140,150), KS                                           JCB00280
  130 D1 = AJ(2,2)*AJ(3,3)-AJ(2,3)*AJ(3,2)                              JCB00290
      D2 = AJ(2,3)*AJ(3,1)-AJ(2,1)*AJ(3,3)                              JCB00300
      D3 = AJ(2,1)*AJ(3,2)-AJ(2,2)*AJ(3,1)                              JCB00310
      DTJ = DSQRT(D1*D1+D2*D2+D3*D3)                                    JCB00320
      GO TO 160                                                         JCB00330
  140 D1 = AJ(1,1)*AJ(3,3)-AJ(1,3)*AJ(3,1)                              JCB00340
      D2 = AJ(1,3)*AJ(3,2)-AJ(1,2)*AJ(3,3)                              JCB00350
      D3 = AJ(1,2)*AJ(3,1)-AJ(1,1)*AJ(3,2)                              JCB00360
      DTJ = DSQRT(D1*D1+D2*D2+D3*D3)                                    JCB00370
      GO TO 160                                                         JCB00380
  150 D1 = AJ(1,1)*AJ(2,2)-AJ(1,2)*AJ(2,1)                              JCB00390
      D2 = AJ(1,2)*AJ(2,3)-AJ(1,3)*AJ(2,2)                              JCB00400
      D3 = AJ(1,3)*AJ(2,1)-AJ(1,1)*AJ(2,3)                              JCB00410
      DTJ = DSQRT(D1*D1+D2*D2+D3*D3)                                    JCB00420
  160 RETURN                                                            JCB00430
      END                                                               JCB00440
C
C***********************************************************************
C***********************************************************************
C
      SUBROUTINE BNSN (A,N,APZRO,ABIGN)                                 BNS00010
C
C     THIS SUBROUTINE CALCULATES A BIG NUMBER (ABIGN) AND THE
C     APPROXIMATE VALUE OF ZERO (APZRO)
C
      IMPLICIT REAL*8(A-H,O-Z)                                          BNS00020
      DIMENSION A(278,40)                                               BNS00030
      ANB = 0.0D0                                                       BNS00040
      DO 110 I=1,N                                                      BNS00050
  110 ANB = ANB+DABS(A(I,1))                                            BNS00060
      APZRO = ANB*1.0D-15/DFLOAT(N)                                     BNS00070
      ABIGN = APZRO*1.0D30                                              BNS00080
      RETURN                                                            BNS00090
      END                                                               BNS00100
C
C***********************************************************************
C***********************************************************************
C
      SUBROUTINE MULT (A,B,C,N,M)                                       MUL00010
C
```

```
C      THIS SUBROUTINE FORMS C=A*B
C
       IMPLICIT REAL*8(A-H,O-Z)                                        MUL00020
       DIMENSION A(278,40), B(278), C(278)                             MUL00030
       DO 120 I=1,N                                                    MUL00040
       IST = I-M-1                                                     MUL00050
       C(I) = B(I)*A(I,1)                                              MUL00060
       DO 120 J=2,M                                                    MUL00070
       IROW = IST+J                                                    MUL00080
       IF (IROW.LE.0) GO TO 110                                        MUL00090
       ICOL = M-J+2                                                    MUL00100
       C(I) = C(I)+A(IROW,ICOL)*B(IROW)                                MUL00110
   110 IR = I+J-1                                                      MUL00120
       IF (IR.GT.N) GO TO 120                                          MUL00130
       C(I) = C(I)+A(I,J)*B(IR)                                        MUL00140
   120 CONTINUE                                                        MUL00150
       RETURN                                                          MUL00160
       END                                                             MUL00170
C
C***********************************************************************
C***********************************************************************
C
       SUBROUTINE LDLT (A,N,M,ASN)                                     LDT00010
C
C      THIS SUBROUTINE DECOMPOSES THE COEFFICIENT MATRIX, CODED BY GILLES
C      CANTIN, NAVAL POSTGRADUATE SCHOOL, MAY, 1972
C
       IMPLICIT REAL*8(A-H,O-Z)                                        LDT00020
       DIMENSION A(278,40)                                             LDT00030
       AVSN = ASN*ASN                                                  LDT00040
       NM = N-1                                                        LDT00050
       DO 130 I=1,NM                                                   LDT00060
       DIAG = A(I,1)                                                   LDT00070
       DO 110 J=2,M                                                    LDT00080
       IF (DABS(A(I,J)).LE.AVSN) A(I,J)=0.0D0                          LDT00090
       A(I,J) = A(I,J)/DIAG                                            LDT00100
   110 CONTINUE                                                        LDT00110
       DO 130 J=2,M                                                    LDT00120
       L = I+J-1                                                       LDT00130
       IF (L.GT.N) GO TO 130                                           LDT00140
       IF (DABS(A(I,J)).LE.AVSN) GO TO 130                             LDT00150
       AA = A(I,J)*DIAG                                                LDT00160
       DO 120 K=J,M                                                    LDT00170
       ML = 1+K-J                                                      LDT00180
   120 A(L,ML) = A(L,ML)-AA*A(I,K)                                     LDT00190
   130 CONTINUE                                                        LDT00200
       RETURN                                                          LDT00210
C
```

```
C     THIS ENTRY POINT PERFORMS FORWARD SUBSTITUTION
C
      ENTRY SLV(Z,B,N,M)                                                LDT00220
      DIMENSION Z(278,40), B(278)                                       LDT00230
      NM = N-1                                                          LDT00240
      DO 140 I=1,NM                                                     LDT00250
      BI = B(I)                                                         LDT00260
      DO 140 J=2,M                                                      LDT00270
      L = I+J-1                                                         LDT00280
      IF (L.GT.N) GO TO 140                                             LDT00290
      B(L) = B(L)-Z(I,J)*BI                                             LDT00300
  140 CONTINUE                                                          LDT00310
      DO 150 I=1,N                                                      LDT00320
  150 B(I) = B(I)/Z(I,1)                                                LDT00330
      DO 160 L=2,N                                                      LDT00340
      IR = N-L+1                                                        LDT00350
      BIRP = B(IR+1)                                                    LDT00360
      DO 160 J=2,M                                                      LDT00370
      IRO = IR-J+2                                                      LDT00380
      IF (IRO.LE.0) GO TO 160                                           LDT00390
      B(IRO) = B(IRO)-Z(IRO,J)*BIRP                                     LDT00400
  160 CONTINUE                                                          LDT00410
      RETURN                                                            LDT00420
      END                                                               LDT00430
C
C***********************************************************************
C***********************************************************************
C
      SUBROUTINE MARK (C,K,M)                                           MRK00010
C
C     THIS SUBROUTINE CONVERTS CYLINDRICAL OR SPHERICAL COORDINATES TO
C     RECTANGULAR COORDINATES.
C
      IMPLICIT REAL*8(A-H,O-Z)                                          MRK00020
      DIMENSION C(278,40)                                               MRK00030
      PI = 3.14159265358979300                                          MRK00040
      IF (K.NE.2) GO TO 120                                             MRK00050
      DO 110 I=1,M                                                      MRK00060
      ANG = C(I,2)*PI/1.8D2                                             MRK00070
      R = C(I,1)                                                        MRK00080
      C(I,1) = R*DCOS(ANG)                                              MRK00090
  110 C(I,2) = R*DSIN(ANG)                                              MRK00100
      WRITE (6,160)                                                     MRK00110
      GO TO 140                                                         MRK00120
  120 DO 130 I=1,M                                                      MRK00130
      ANGZ = C(I,3)*PI/1.8D2                                            MRK00140
      ANGX = C(I,2)*PI/1.8D2                                            MRK00150
      R = C(I,1)                                                        MRK00160
```

```
      SIN = DSIN(ANGZ)                                                  MRK00170
      C(I,1) = R*SIN*DCOS(ANGX)                                         MRK00180
      C(I,2) = R*SIN*DSIN(ANGX)                                         MRK00190
  130 C(I,3) = R*DCOS(ANGZ)                                             MRK00200
      WRITE (6,170)                                                     MRK00210
  140 DO 150 J=1,M                                                      MRK00220
  150 WRITE (6,180) J,(C(J,L),L=1,3)                                    MRK00230
      RETURN                                                            MRK00240
  160 FORMAT ('0',5X,'COORDINATES GIVEN IN CYLINDRICAL SYSTEM OF REFERENMRK00250
     1CE',//,6X,'RECTANGULAR COORDINATES FOLLOW',///,5X,' NODE NUMBER',1MRK00260
     20X,'X COORDINATE',12X,'Y COORDINATE',12X,'Z COORDINATE',///)     MRK00270
  170 FORMAT ('0',5X,'COORDINATES GIVEN IN SPHERICAL SYSTEM OF REFERENCEMRK00280
     1',//,6X,'RECTANGULAR COORDINATES FOLLOW',///,5X,' NODE NUMBER',10XMRK00290
     2,'X COORDINATE',12X,'Y COORDINATE',12X,'Z COORDINATE',///)       MRK00300
  180 FORMAT (10X,I5,5X,1P3G24.16)                                      MRK00310
      END                                                               MRK00320
C
C***********************************************************************
C***********************************************************************
C
      SUBROUTINE PUNC(A,WHEN)                                           PUN00010
C
C     THIS SUBROUTINE PUNCHES REQUESTED OUTPUT
C
      IMPLICIT REAL*8(A-H,O-Z)                                          PUN00020
      COMMON /MTRX/ CORD(32,3),ENN(32,32),DCOL(32,3),COL(32),AJ(3,3),AJIPUN00030
     1(3,3),ENW(32,32),FE(32),PROP(278,4),COARD(278,3),F(278),PHI(278),PPUN00040
     2HIDOT(278),PHIWRK(278),PHIDDT(278),BGH(278,40),BGC(278,40),PROB(27PUN00050
     38,3),PUNCH(20)                                                    PUN00060
      COMMON /FLP/ DTJ,AKX,AKY,AKZ,FACT,PHIREF,APZRO,ABIGN,TIM,DT,ENDT  PUN00070
      COMMON /INT/ IGO,NBAND,NNODE,NMAT,NEL,NPEL,NGP,NELBC,NNBC,IPROB,KCPUN00080
     1,NTSC,IPUN,IPRT,ISTP                                              PUN00090
      DIMENSION A(278)                                                  PUN00100
      DATA IFLAG/1/                                                     PUN00110
      IF (IPROB.EQ.1) GO TO 120                                         PUN00120
      TEST = PUNCH(IFLAG)-WHEN                                          PUN00130
      IF (DABS(TEST).LE.1.0D-06) GO TO 110                              PUN00140
      RETURN                                                            PUN00150
  110 WRITE (7,140) WHEN                                                PUN00160
  120 DO 130 I=1,NNODE                                                  PUN00170
      AP = A(I)+PHIREF                                                  PUN00180
  130 WRITE (7,150) I,AP                                                PUN00190
      IFLAG = IFLAG+1                                                   PUN00200
      RETURN                                                            PUN00210
  140 FORMAT (F10.5)                                                    PUN00220
  150 FORMAT (I5,F9.3)                                                  PUN00230
      END                                                               PUN00240
```

APPENDIX G

COMPUTER LISTING FOR STRESS PROGRAM

```
C     ********************************************************************CNTP0010
C     THIS PROGRAM PERFORMS A STRESS AND STRAIN ANALYSIS OF THREE          CNTP0020
C     DIMENSIONAL OBJECTS USING NUMERICALLY INTEGRATED ISOPARAMETRIC       CNTP0030
C     FINITE ELEMENTS. THIS PARTICULAR VERSION USES TWENTY (20) NODAL      CNTP0040
C     POINT ELEMENTS ALLOWING TRIQUADRATIC VARIATION OF THE DISPLACEMENT   CNTP0050
C     FUNCTIONS THROUGHOUT THE ELEMENTS.                                   CNTP0060
C     PROFESSOR GILLES CANTIN, DECEMBER 1970, REVISED IN 1971 BY           CNTP0070
C     LCDR. LEONIDAS , NAVAL POSTGRADUATE SCHOOL, UPDATED IN 1975 BY       CNTP0071
C     LT. BUBECK, NAVAL POSTGRADUATE SCHOOL, TO INCLUDE TRANSIENT THERMAL  CNTP0072
C     STRESSES.                                                            CNTP0080
C     ********************************************************************CNTP0090
C                                                                          CNTP0100
C     INPUT CARDS NEEDED                                                   CNTP0110
C                                                                          CNTP0120
C    INITIALIZATION CARD(1A8,I5) WITH THE WORD START IN THE FIRST 8        CNTP0130
C    COLUMNS, PLUS AN INTEGER SPECIFYING THE NUMBER OF SETS OF LAOD        CNTP0140
C    CONDITIONS( MAX OF 10)                                                CNTP0150
C                                                                          CNTP0160
C    PROBLEM IDENTIFICATION CARD (10A8) , ANY TITLE CAN BE USED            CNTP0170
C                                                                          CNTP0180
C    PARAMETER CARD (16I5),ONE CARD PER PROBLEM                            CNTP0190
C    COL.      VARIABLE                                                    CNTP0200
C     1-5      NEL       TOTAL NUMBER OF ELEMENTS                          CNTP0210
C     6-10     NDPT      TOTAL NUMBER OF NODAL POINTS                      CNTP0220
C    11-15     NS        BLOCK SIZE USED IN THE EQUATION SOLVER            CNTP0230
C    16-20     NPEL      NUMBER OF NODES PER ELEMENT                       CNTP0240
C    21-25     NMAT      NUMBER OF MATERIALS(MAX.=10)                      CNTP0250
C                        IF NODAL PROPERTIES, NMAT=-1                      CNTP0260
C    26-30     NCLD      TOTAL NUMBER OF NODES WITH CONCENTRATED LOADS     CNTP0270
C    31-35     NPBC      TOTAL NUMBER OF NODES WITH BOUNDARY CONDITIONS    CNTP0280
C    36-40     NPBC2     NUMBER OF NODES WITH PREDEFINED DISPLACEMENTS     CNTP0290
C    41-45     NLINE     NUMBER OF LINES PER OUTPUT PAGE DESIRED FOR THE   CNTP0300
C                        TABLE OF JOINT STRESSES AND STRAINS               CNTP0310
C    46-50     KEL       IF THIS IS 0 OR BLANK ELEMENT STRESSES AND        CNTP0320
C    51-55     NTLD      THERMAL LOAD PRESENT                              CNTP0330
C    56-60     NBLD      BODY FORCE PRESENT                                CNTP0340
C    61-65     NSLD      INITIAL STRETCH LOAD PRESENT                      CNTP0350
C    66-70     NREC7     NUMBER OF RECORDS ON UNIT SEVEN(4/6 FOR QUADS/CUBICS)CNTP0360
C    71-75     NGPS      NUMBER OF GAUSS POINTS FOR STIFFNESS INTEGRATION  CNTP0370
C                        IF NGPS=0, A DATA CHECK IS PERFORMED.             CNTP0380
C    76-80     NGPL      NUMBER OF GAUSS POINTS FOR LOAD GENERATION        CNTP0390
C                        STRAINS WILL NOT BE PRINTED, IF THIS IS 1 A PRINTING CNTP0400
C                        RESULTS.  THIS PRONTING CAN BE VOLUMINOUS.        CNTP0410
```

```
C                                                                               CNTP0420
C   * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *  CNTP0430
C                                                                               CNTP0440
C                  N-O-T-A  B-E-N-E                                             CNTP0450
C                                                                               CNTP0460
C   FOR THE REST OF THE INPUT CARDS, OBJECT TIME FORMATS ARE USED              CNTP0470
C   THROUGHOUT, UNFAMILIAR USERS SHOULD LEARN ABOUT THIS FEATURE FROM          CNTP0480
C   ANY GOOD STANDARD BOOK ON FORTRAN 4                                         CNTP0490
C                                                                               CNTP0500
C   * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *  CNTP0510
C                                                                               CNTP0520
C***ELEMENT CARD DECK                                                           CNTP0530
C                                                                               CNTP0540
C        TITLE CARD  (10A8)  ANY TITLE IN COLUMNS 2-80                          CNTP0550
C                                                                               CNTP0560
C        FORMAT CARD (10A8)  FORMAT SELECTED FOR THE CONNECTIVITY CARDS         CNTP0570
C                            THE SAME FORMAT WILL BE USED FOR THE ECHO          CNTP0580
C                            CHECK. (SAME FOR ALL SUBSEQUENT DECKS)             CNTP0590
C                            THE FORMAT MUST ACCOMODATE 20 OR 21                CNTP0600
C                            INTEGERS AS DESCRIBED NEXT                         CNTP0610
C                                                                               CNTP0620
C        CONNECTIVITY CARDS (NEL) PER DECK, CONTAINING THE STANDARD             CNTP0630
C        SEQUENCE OF TWENTY NODES PER ELEMENT FOLLOWED BY A MATERIAL            CNTP0640
C        IDENTIFICATION NUMBER IF NEEDED                                        CNTP0650
C                                                                               CNTP0660
C***JOINT CARD DECK                                                             CNTP0670
C                                                                               CNTP0680
C        TITLE CARD  (10A8)  ANY TITLE IN COLUMNS 2-80                          CNTP0690
C                                                                               CNTP0700
C        FORMAT CARD (10A8)  FORMAT SELECTED FOR THE JOINT CARDS                CNTP0710
C                            (1 INTEGER, 3 FL. PT, 1 INTEGER)                   CNTP0720
C                                                                               CNTP0730
C        COORDINATE CARDS (NDPT) PER DECK CONTAINING FIRST AN INTEGER           CNTP0740
C        IDENTIFICATION NUMBER FOR THE JOINT FOLLOWED BY EITHER THREE           CNTP0750
C        CARTESIAN COORDINATES (X,Y,Z) OR THREE CYLINDRICAL COORDINATES         CNTP0760
C        (R,PHI,Z) FOLLOWED BY AN INTEGER DIFFERENT FROM ZERO IF                CNTP0770
C        CYLINDRICAL COORDINATES ARE USED                                       CNTP0780
C                                                                               CNTP0790
C***MATERIAL PROPERTY TABLE                                                     CNTP0800
C                                                                               CNTP0810
C        IF NODAL MATERIAL PROPERTIES(NMAT=-1) ARE USED REFER TO (2)            CNTP0820
C   (1)  TITLE CARD  (10A8)  ANY TITLE IN COLUMNS 2-80                          CNTP0830
C                                                                               CNTP0840
C        FORMAT CARD (10A8)  FORMAT SELECTED FOR THE MATERIAL PROPERTIESCNTP0850
C                            (1 INTEGER, 3 FL. POINT NUMBERS)                   CNTP0860
C                                                                               CNTP0870
C        MATERIAL PROPERTY (NMAT) CARDS, MUST BE LESS THAN 10                   CNTP0880
C        CONTAINING AN INTEGER IDENTIFICATION NUMBER FOR THE MATERIAL           CNTP0890
```

```
C        FOLLOWED BY THE VALUES OF YOUNG'S MODULUS, POISSON'S RATIO      CNTP0900
C        AND COEFFICIENT OF THERMAL EXPANSION                            CNTP0910
C                                                                        CNTP0920
C   (2)  TITLE CARD  (10A8)  ANY TITLE IN COLUMNS 2-80                   CNTP0930
C                                                                        CNTP0940
C        FORMAT CARD (10A8) FORMAT SELECTED FOR NODAL MATERIAL           CNTP0950
C        PROPERTIES(2 INTEGERS, 3 FL. POINT NUMBERS)                     CNTP0960
C                                                                        CNTP0970
C        EACH NODE MUST HAVE PROPERTIES ASSIGNED TO IT.  IN ORDER TO     CNTP0980
C        REDUCE THE NUMBER OF INPUT CARDS, IF ALL NODES(CONSECUTIVE)     CNTP0990
C        BETWEEN NODE "N" AND NODE "M" ARE THE SAME, THE TWO INTEGERS    CNTP1000
C        SPECIFIED ARE "N" AND "M".  IF NODE "N" IS UNIQUE IN ITS        CNTP1010
C        PROPERTIES THE BOTH INTEGERS ARE "N".  THE FLOATING POINT       CNTP1020
C        NUMBERS ARE YOUNG'S MODULUS, POISSON RATIO, AND THE COEFFICIENTCNTP1030
C        OF THERMAL EXPANSION.                                           CNTP1040
C                                                                        CNTP1050
C***BOUNDARY CONDITION DECK                                              CNTP1060
C                                                                        CNTP1070
C        TITLE CARD  (10A8)  ANY TITLE IN COLUMNS 2-80                   CNTP1080
C                                                                        CNTP1090
C        FORMAT CARD (10A8)  FORMAT SELECTED FOR THE BOUNDARY CARDS      CNTP1100
C                            (4 INTEGERS)                                CNTP1110
C                                                                        CNTP1120
C        BOUNDARY CARD (NPBC) PER DECK CONTAINING FIRST AN INTEGER       CNTP1130
C        IDENTIFICATION FOR THE NODE FOLLOWED BY A  0  FOR A FREE NODE   CNTP1140
C        FOLLOWED BY  1  FOR A ZERO DISPLACEMENT CONSTRAINT, FOLLOWED    CNTP1150
C        BY  A  2  FOR A PREDEFINED DISPLACEMENT.  CARDS MUST BE         CNTP1160
C        SEQUENCED IN ASCENDING ORDER OF NODE NUMBER                     CNTP1170
C                                                                        CNTP1180
C***PREDEFINED DISPLACEMENT DECK (OMIT THIS DECK IF NPBC2 IS ZERO)       CNTP1190
C                                                                        CNTP1200
C        TITLE CARD  (10A8)  ANY TITLE IN COLUMNS 2-80                   CNTP1210
C                                                                        CNTP1220
C        FORMAT CARD (10A8) FORMAT SELECTED FOR THE DISPLACEMENT CARDS   CNTP1230
C                            (1 INTEGER, 3 FL. POINT NUMBERS)            CNTP1240
C                                                                        CNTP1250
C        SPECIFIED DISPLACEMENTS (NPBC2) PER DECK CONTAINING AN INTEGER  CNTP1260
C        IDENTIFICATION NUMBER FOR THE NODE FOLLOWED BY THE SPECIFIED,   CNTP1270
C        DISPLACEMENT COMPONENTS IN THE GLOBAL COORDINATE SYSTEM (X,Y,Z)CNTP1280
C                                                                        CNTP1290
C***CONCENTRATED LOAD CARD DECK(OMIT IF NCLD=0)                          CNTP1300
C                                                                        CNTP1310
C        TITLE CARD  (10A8)  ANY TITLE IN COLUMNS 2-80                   CNTP1320
C                                                                        CNTP1330
C        FORMAT CARD (10A8)  FORMAT SELECTED FOR THE LOAD CARDS          CNTP1340
C                            (1 INTEGER, 3 FL. POINT NUMBERS)            CNTP1350
C                                                                        CNTP1360
C        LOAD CARDS (NCLD) CARDS PER DECK, CONTAINING AN INTEGER         CNTP1370
```

```
C        IDENTIFICATION FOLLOWED BY THE VALUES OF THE COMPONENTS OF THE
C        JOINT LOAD IN THE GLOBAL SYSTEM OF REFERENCE (X,Y,Z)
C        LOAD CARDS MUST BE IN ASCENDING ORDER OF THE JOINT NUMBER
C
C
C***NLOAD SETS OF LOAD CONDITIONS ARE READ AS DESCRIBED.
C
C***THERMAL LOAD CONDITIONS(OMIT IF NTLD=0)
C
C        TITLE CARD(10A8)  ANY TITLE INCOLUMNS 2-80
C
C        FORMAT CARD(10A8) FORMAT SELECTED FOR NODAL TEMPERATURES
C                          (1 INTEGER,1 FL. POINT NUMBER)
C
C        SPECIFY NODAL NUMBER AND TEMPERATURE, SEQUENCED IN ORDER OF
C        ASCENDING NODAL NUMBER; NNODE CARDS REQUIRED.
C
C        IF NTLD=-1 , TEMPERATURES ARE READ FROM FILE 33.
C
C***BODY FORCE LOAD CONDITIONS(OMIT IF NBLD=0)
C
C        TITLE CARD(10A8) ANY TITLE IN COLUMNS 2-80
C
C        FORMAT CARD(10A8) FORMAT SELECTED FOR BODY FORCES
C                          (1 INTEGER, 3 FL. POINT NUMBERS)
C
C        SPECIFY NODAL NUMBER AND X,Y,Z COMPONENTS OF BODY FORCE,
C        SEQUENCED IN ORDER OF ASCENDING NODAL NUMBERS, NNODE CARDS.
C
C        IF NBLD=-1 , BODY FORCES ARE READ FROM FILE 34.
C
C***INITIAL STRETCH LOAD CONDITIONS(OMIT IF NSLD=0)
C
C        TITLE CARD(10A8) ANY TITLE IN COLUMNS 2-80
C
C        FORMAT CARD(10A8) FORMAT SELECTED FOR INITIAL STRETCH CONDITION
C                          (1 INTEGER, 3 FL. POINT NUMBERS)
C
C        SPECIFY NODAL NUMBER AND X,Y,Z COMPONENTS OF INITIAL STRETCH
C        SEQUENCED IN ORDER OF ASCENDING NODAL NUMBER, NNODE CARDS
C
C        IF NSLD=-1 , STRETCH CONDTIONS ARE READ FROM FILE 35.
C
C
C
C        IF ANOTHER PROBLEM FOLLOWS PLACE COMPLETE DECK WITH INITIAL
C        "START" CARD.  IF END OF RUN PLACE A CARD WITH "STOP" IN FIRST
C        8 COLUMNS.
```

```
C
      ******************************************************************
      *                                                                *
      *                        CONTROL PROGRAM                         *
      *                                                                *
      ******************************************************************
      IMPLICIT REAL*8(A-H,O-Z)
      COMMON /NB1/NEL,NDPT,NPEL,NDF,NEQ,NN,MM,NS,NCOUNT,NST,NSTF,NCLD,
     1NPBC,NPBC2,NBCL,NBCH,NMAT,NCON21,NDFP,NLINE,NREC7,KEL,NTLD,NBLD,
     2NSLD
      COMMON /IFILE/ I7,I8,I10,I11,I12,I13,I18,I20,I21,I25,I26,I29
      DEFINE FILE 7( 120,1800,U,I7),8( 100,1800,U,I8),
     1           10( 200,    6,U,I10),11( 200 ,    6,U,I11),
     2           12(2000,   14,U,I12),13( 200,   14,U,I13),
     3           18(2000,    2,U,I18),20( 400,  120,U,I20),
     4           21( 800,    6,U,I21),29(2000,    6,U,I29)
C
C           INITIALIZATION OF THE PROGRAM
C
   10 CALL MAINE
      GO TO 10
      END
C
C**********************************************************************
C**********************************************************************
C
      SUBROUTINE MAINE
C
C     SUBROUTINE MAINE PERFORMS ALL CALLING OPERATIONS ON THE VARIOUS
C     SUBPROGRAMS
C
      IMPLICIT REAL*8(A-H,O-Z)
      COMMON /NB1/ NEL,NDPT,NPEL,NDF,NEQ,NN,MM,NS,NCOUNT,NST,NSTF,NCLD,N
     1PBC,NPBC2,NBCL,NBCH,NMAT,NCON21,NDFP,NLINE,NREC7,KEL,NTLD,NBLD,NSL
     2D
      COMMON /B1/ ELDAT(10,3),CLOAD(4),BCON(3)
      COMMON /B12/ ELDATN(3),BYNP(3),STNP(3),THNP,FLOAD(3)
      COMMON /B2/ ELAST(6,6),COORD(3),COREL(20,3),UJNT(3),SSSS1(7),SSNN1
     1(7),ALPHA
      COMMON /B3/ AK1(60,60),AK2(60,60),AK3(60,60),RB1(60),RB2(60),RB3(6
     10)
      COMMON /GAUSS/ NGPS,NGPL
      COMMON /CNTRL/ NLOAD
      DIMENSION SSSS(6), SSNN(6), ELDATE(20,3), TEEL(20)
      EQUIVALENCE (SSSS(1),SSSS1(1)), (SSNN(1),SSNN1(1)), (AK3(1,34),TEE
     1L(1)), (RB2(1),ELDATE(1,1))
      NIRK(I,J) = (I-1)*NDPT+J
C
```

```
C
C     INPUT ALL DATA
C
      CALL SETIME
      CLOCK = ITIME(0)*0.01
      CALL INPUT
      CALL GETIME (IET)
      CPUTM = IET*0.000026
      WRITE (6,440)
      WRITE (6,430)
      CLOCK = ITIME(0)*0.01-CLOCK
      WRITE (6,520) CLOCK,CPUTM
      IF (NGPS.EQ.0) RETURN
C
C     FORM ELEMENT NODAL ARRAYS BY CALLING SORT
C
      CALL SETIME
      CLOCK = ITIME(0)*0.01
      CALL SORT
      CALL GETIME (IET)
      CPUTM = IET*0.000026
      WRITE (6,450)
      CLOCK = ITIME(0)*0.01-CLOCK
      WRITE (6,520) CLOCK,CPUTM
C
C     FORM STIFFNESS MATRICES
C
      CALL SETIME
      CLOCK = ITIME(0)*0.01
      CALL FSTF
      CALL GETIME (IET)
      CPUTM = IET*0.000026
      WRITE (6,460)
      CLOCK = ITIME(0)*0.01-CLOCK
      WRITE (6,520) CLOCK,CPUTM
C
C     FORM TOTAL STIFFNESS NATRIX IN BLOCKS
C
      CALL SETIME
      CLOCK = ITIME(0)*0.01
      CALL MERGE
      CALL GETIME (IET)
      CPUTM = IET*0.000026
      WRITE (6,470)
      CLOCK = ITIME(0)*0.01-CLOCK
      WRITE (6,520) CLOCK,CPUTM
      IF (IABS(NTLD)+IABS(NBLD)+IABS(NSLD).EQ.0) GO TO 10
C
C     FORM LOAD VECTOR BY LOAD GENERATION
```

```
C
      CALL SETIME                                                       SMA00530
      CLOCK = ITIME(0)*0.01                                             SMA00540
      CALL LODGEN                                                       SMA00550
      CALL GETIME (IET)                                                 SMA00560
      CPUTM = IET*0.000026                                              SMA00570
      WRITE (6,440)                                                     SMA00580
      WRITE (6,480)                                                     SMA00590
      CLOCK = ITIME(0)*0.01-CLOCK                                       SMA00600
      WRITE (6,520) CLOCK,CPUTM                                         SMA00610
C
C     FORM TOTAL LOAD VECTORS
C
   10 CALL SETIME                                                       SMA00620
      CLOCK = ITIME(0)*0.01                                             SMA00630
      CALL DLOAD                                                        SMA00640
      CALL GETIME (IET)                                                 SMA00650
      CPUTM = IET*0.000026                                              SMA00660
      WRITE (6,490)                                                     SMA00670
      CLOCK = ITIME(0)*0.01-CLOCK                                       SMA00680
      WRITE (6,520) CLOCK,CPUTM                                         SMA00690
      IF (NPBC2.NE.0) GO TO 20                                          SMA00700
C
C     APPLY ANY BOUNDARY CONDITIONS
C
      CALL SETIME                                                       SMA00710
      CLOCK = ITIME(0)*0.01                                             SMA00720
      CALL BCOND                                                        SMA00730
      CALL GETIME (IET)                                                 SMA00740
      CPUTM = IET*0.000026                                              SMA00750
      WRITE (6,500)                                                     SMA00760
      CLOCK = ITIME(0)*0.01-CLOCK                                       SMA00770
      WRITE (6,520) CLOCK,CPUTM                                         SMA00780
      GO TO 30                                                          SMA00790
   20 CALL SETIME                                                       SMA00800
      CLOCK = ITIME(0)*0.01                                             SMA00810
      CALL BCOND2                                                       SMA00820
      CALL GETIME (IET)                                                 SMA00830
      CPUTM = IET*0.000026                                              SMA00840
      WRITE (6,510)                                                     SMA00850
      CLOCK = ITIME(0)*0.01-CLOCK                                       SMA00860
      WRITE (6,520) CLOCK,CPUTM                                         SMA00870
C
C     SOLVE SYSTEM OF EQUATIONS FOR DISPLACEMENTS
C
   30 CALL SETIME                                                       SMA00880
      CLOCK = ITIME(0)*0.01                                             SMA00890
      CALL SOLVE                                                        SMA00900
```

```
      CALL GETIME (IET)                                                 SMA00910
      CPUTM = IET*0.000026                                              SMA00920
      WRITE (6,440)                                                     SMA00930
      WRITE (6,530)                                                     SMA00940
      CLOCK = ITIME(0)*0.01-CLOCK                                       SMA00950
      WRITE (6,520) CLOCK,CPUTM                                         SMA00960
      IF (NTLD.EQ.-1) REWIND 30                                         SMA00970
      IF (NTLD.NE.0) REWIND 24                                          SMA00980
C
C     LOOP ON DIFFERENT LOADS TO COMPUTE ALL STRESSES AND STRAINS
C
      DO 370 IL=1,NLOAD                                                 SMA00990
      IF (NTLD.EQ.-1) READ (30) TIME                                    SMA01000
      WRITE (6,590) IL                                                  SMA01010
      IF (NTLD.EQ.-1) WRITE (6,380) TIME                                SMA01020
      IF (NTLD.EQ.0) GO TO 60                                           SMA01030
      NL1 = 1                                                           SMA01040
      NL2 = NLINE                                                       SMA01050
   40 WRITE (6,390)                                                     SMA01060
      DO 50 I=NL1,NL2                                                   SMA01070
      NTK = NTRK(IL,I)                                                  SMA01080
      CALL RDISKT (NTK)                                                 SMA01090
   50 WRITE (6,600) I,THNP                                              SMA01100
      IF (NL2.EQ.NDPT) GO TO 60                                         SMA01110
      NL1 = NL2+1                                                       SMA01120
      NL2 = NL2+NLINE                                                   SMA01130
      IF (NL2.GT.NDPT) NL2=NDPT                                         SMA01140
      WRITE (6,620)                                                     SMA01150
      GO TO 40                                                          SMA01160
   60 IF (NBLD.EQ.0) GO TO 90                                           SMA01170
      WRITE (6,620)                                                     SMA01180
      NL1 = 1                                                           SMA01190
      NL2 = NLINE                                                       SMA01200
   70 WRITE (6,400)                                                     SMA01210
      DO 80 I=NL1,NL2                                                   SMA01220
      NTK = NTRK(IL,I)                                                  SMA01230
      CALL RDISBF (NTK)                                                 SMA01240
   80 WRITE (6,610) I,BYNP                                              SMA01250
      IF (NL2.EQ.NDPT) GO TO 90                                         SMA01260
      NL1 = NL2+1                                                       SMA01270
      NL2 = NL2+NLINE                                                   SMA01280
      IF (NL2.GT.NDPT) NL2=NDPT                                         SMA01290
      WRITE (6,620)                                                     SMA01300
      GO TO 70                                                          SMA01310
   90 IF (IABS(NTLD)+IABS(NBLD)+IABS(NSLD).EQ.0) GO TO 150              SMA01320
      TLOAD1 = 0.0D0                                                    SMA01330
      TLOAD2 = 0.0D0                                                    SMA01340
      TLOAD3 = 0.0D0                                                    SMA01350
```

```
      NL1 = 1                                                           SMA01360
      NL2 = NLINE                                                       SMA01370
  100 WRITE (6,410) IL                                                  SMA01380
      DO 110 I=NL1,NL2                                                  SMA01390
      NTK = NTRK(IL,I)                                                  SMA01400
      CALL RDISLF (NTK)                                                 SMA01410
      WRITE (6,610) I,FLOAD                                             SMA01420
      TLOAD1 = TLOAD1+FLOAD(1)                                          SMA01430
      TLOAD2 = TLOAD2+FLOAD(2)                                          SMA01440
  110 TLOAD3 = TLOAD3+FLOAD(3)                                          SMA01450
      IF (NL2.EQ.NDPT) GO TO 120                                        SMA01460
      NL1 = NL2+1                                                       SMA01470
      NL2 = NL2+NLINE                                                   SMA01480
      IF (NL2.GT.NDPT) NL2=NDPT                                         SMA01490
      GO TO 100                                                         SMA01500
  120 IF (NCLD.EQ.0) GO TO 140                                          SMA01510
      REWIND 14                                                         SMA01520
      DO 130 I=1,NCLD                                                   SMA01530
      READ (14) CLOAD                                                   SMA01540
      TLOAD1 = TLOAD1+CLOAD(1)                                          SMA01550
      TLOAD2 = TLOAD2+CLOAD(2)                                          SMA01560
  130 TLOAD3 = TLOAD3+CLOAD(3)                                          SMA01570
  140 WRITE (6,420) TLOAD1,TLOAD2,TLOAD3                                SMA01580
C
C     FORM NODAL DISPLACEMENTS AND REACTIONS
C
  150 CALL SETIME                                                       SMA01590
      CLOCK = ITIME(0)*0.01                                             SMA01600
      CALL DISP (IL)                                                    SMA01610
      CALL GETIME (IET)                                                 SMA01620
      CPUTM = IET*0.000026                                              SMA01630
      WRITE (6,440)                                                     SMA01640
      WRITE (6,540)                                                     SMA01650
      CLOCK = ITIME(0)*0.01-CLOCK                                       SMA01660
      WRITE (6,520) CLOCK,CPUTM                                         SMA01670
      CALL SETIME                                                       SMA01680
      CLOCK = ITIME(0)*0.01                                             SMA01690
      REWIND 9                                                          SMA01700
      REWIND 19                                                         SMA01710
      DO 160 I=1,7                                                      SMA01720
      SSNN1(I) = 0.0D0                                                  SMA01730
  160 SSSS1(I) = 0.0D0                                                  SMA01740
      DO 170 I=1,NDPT                                                   SMA01750
      IS = (IL-1)*NDPT+I                                                SMA01760
      CALL WDISKS (IS)                                                  SMA01770
  170 CALL WDISKN (I)                                                   SMA01780
      IF (NMAT.EQ.1) GO TO 200                                          SMA01790
      IF (NMAT.EQ.-1) GO TO 220                                         SMA01800
```

```
      NMAT1 = 0                                                         SMA01810
      REWIND 16                                                         SMA01820
      DO 190 I=1,NEL                                                    SMA01830
      READ (19) COREL                                                   SMA01840
      READ (16) NCON21                                                  SMA01850
      IF (NTLD.NE.0) READ (24) TEEL                                     SMA01860
      IF (NMAT1.EQ.NCON21) GO TO 180                                    SMA01870
      NMAT1 = NCON21                                                    SMA01880
      ALPHA = ELDAT(NCON21,3)                                           SMA01890
      CALL ELPROP (NMAT1)                                               SMA01900
  180 CALL STRESS (I,IL)                                                SMA01910
  190 CONTINUE                                                          SMA01920
      GO TO 240                                                         SMA01930
  200 ALPHA = ELDAT(1,3)                                                SMA01940
      DO 210 I=1,NEL                                                    SMA01950
      IF (NTLD.NE.0) READ (24) TEEL                                     SMA01960
      READ (19) COREL                                                   SMA01970
  210 CALL STRESS (I,IL)                                                SMA01980
      GO TO 240                                                         SMA01990
  220 REWIND 23                                                         SMA02000
      DO 230 I=1,NEL                                                    SMA02010
      READ (19) COREL                                                   SMA02020
      READ (23) ELDATE                                                  SMA02030
      IF (NTLD.NE.0) READ (24) TEEL                                     SMA02040
  230 CALL STRESS (I,IL)                                                SMA02050
  240 NPAGE = NDPT/NLINE                                                SMA02060
      NL = NPAGE*NLINE                                                  SMA02070
      NL1 = NL+1                                                        SMA02080
      NLAST = NDPT-NL                                                   SMA02090
      IF (NPAGE.EQ.0) GO TO 270                                         SMA02100
      DO 260 I=1,NPAGE                                                  SMA02110
      WRITE (6,580)                                                     SMA02120
      I1 = (I-1)*NLINE+1                                                SMA02130
      I2 = I1+NLINE-1                                                   SMA02140
      DO 260 J=I1,I2                                                    SMA02150
      JS = (IL-1)*NDPT+J                                                SMA02160
      CALL RDISKS (JS)                                                  SMA02170
      DO 250 K=1,6                                                      SMA02180
  250 SSSS1(K) = SSSS1(K)/SSSS1(7)                                      SMA02190
      CALL PRNSTR (SSSS1,S1,S2,S3)                                      SMA02200
  260 WRITE (6,570) J,SSSS,S1,S2,S3                                     SMA02210
      IF (NLAST.EQ.0) GO TO 300                                         SMA02220
  270 WRITE (6,580)                                                     SMA02230
      DO 290 I=NL1,NDPT                                                 SMA02240
      IS = (IL-1)*NDPT+I                                                SMA02250
      CALL RDISKS (IS)                                                  SMA02260
      DO 280 K=1,6                                                      SMA02270
  280 SSSS1(K) = SSSS1(K)/SSSS1(7)                                      SMA02280
```

```
      CALL PRNSTR (SSSS1,S1,S2,S3)                                      SMA02290
  290 WRITE (6,570) I,SSSS,S1,S2,S3                                     SMA02300
  300 CONTINUE                                                          SMA02310
      IF (NPAGE.EQ.0) GO TO 330                                         SMA02320
      DO 320 I=1,NPAGE                                                  SMA02330
      WRITE (6,560)                                                     SMA02340
      I1 = (I-1)*NLINE+1                                                SMA02350
      I2 = I1+NLINE-1                                                   SMA02360
      DO 320 J=I1,I2                                                    SMA02370
      CALL RDISKN (J)                                                   SMA02380
      DO 310 K=1,6                                                      SMA02390
  310 SSNN1(K) = SSNN1(K)/SSNN1(7)                                      SMA02400
  320 WRITE (6,570) J,SSNN                                              SMA02410
      IF (NLAST.EQ.0) GO TO 360                                         SMA02420
  330 WRITE (6,560)                                                     SMA02430
      DO 350 I=NL1,NDPT                                                 SMA02440
      CALL RDISKN (I)                                                   SMA02450
      DO 340 K=1,6                                                      SMA02460
  340 SSNN1(K) = SSNN1(K)/SSNN1(7)                                      SMA02470
  350 WRITE (6,570) I,SSNN                                              SMA02480
  360 CONTINUE                                                          SMA02490
      CALL GETIME (IET)                                                 SMA02500
      CPUTM = IET*0.000026                                              SMA02510
      WRITE (6,440)                                                     SMA02520
      WRITE (6,550)                                                     SMA02530
      CLOCK = ITIME(0)*0.01-CLOCK                                       SMA02540
      WRITE (6,520) CLOCK,CPUTM                                         SMA02550
  370 CONTINUE                                                          SMA02560
      RETURN                                                            SMA02570
  380 FORMAT ('0',14X,' TIME OF LOAD VECTOR = ',F10.5)                  SMA02580
  390 FORMAT ('0',14X,' NODAL TEMPERATURES',//,14X,'NODE',6X,'TEMPERATURSMA02590
     1E'/)                                                              SMA02600
  400 FORMAT ('0',14X,'NODAL BODY FORCES',//,14X,'NODE',6X,'BODY FORCE')SMA02610
  410 FORMAT ('1',25X,' CONSISTENT LOAD VECTOR NO.',I5,//,' ',25X,'X',25SMA02620
     1X,'Y',25X,'Z',/)                                                  SMA02630
  420 FORMAT ('0','TOTAL LOADS ARE ',1P3D25.16)                         SMA02640
  430 FORMAT (5X,'***** INPUT *****')                                   SMA02650
  440 FORMAT (///)                                                      SMA02660
  450 FORMAT (5X,'***** SORT *****')                                    SMA02670
  460 FORMAT (5X,'***** FSTF *****')                                    SMA02680
  470 FORMAT (5X,'***** MERGE *****')                                   SMA02690
  480 FORMAT (5X,'*****LODGEN*****')                                    SMA02700
  490 FORMAT (5X,'***** DLOAD *****')                                   SMA02710
  500 FORMAT (5X,'***** BCOND *****')                                   SMA02720
  510 FORMAT (5X,'***** BCOND2 *****')                                  SMA02730
  520 FORMAT (5X,7H TIME =,1F7.2,8H SECONDS,6H(CPU =,1F7.2,9H SECONDS)//SMA02740
     1)                                                                 SMA02750
  530 FORMAT (5X,'***** SOLVE *****')                                   SMA02760
```

```
  540 FORMAT (5X,'***** DISP *****')                                         SMA02770
  550 FORMAT (5X,'***** STRESS *****')                                       SMA02780
  560 FORMAT (1H1,//,' A-V-E-R-A-G-E  S-T-R-A-I-N-S  AT  THE  JOINTS ',     SMA02790
     1 //,2X,'JOINT',                                                        SMA02800
     2 9X,'SNX',12X,'SNY',12X,'SNZ',12X,'SNXY',11X,'SNYZ',11X,'SNZX')        SMA02810
  570 FORMAT (2X,I4,2X,6(1PD15.6),1X,3(1PD11.3))                            SMA02820
  580 FORMAT (1H1,//,' A-V-E-R-A-G-E  S-T-R-E-S-S-E-S  AT  THE  JOINTS 'SMA02830
     1,//,2X,'JOINT',                                                        SMA02840
     2 7X,'SSX',12X,'SSY',12X,'SSZ',12X,'SSXY',11X,'SSYZ',11X,'SSZX',10XSMA02850
     3,'S1',9X,'S2',9X,'S3',//)                                              SMA02860
  590 FORMAT ('1',120('*'),//,45X,'SOLUTION FOR LOAD VECTOR NO.',I5,//,  SMA02870
     1 120('*'))                                                             SMA02880
  600 FORMAT (' ',I16,8X,F10.5)                                              SMA02890
  610 FORMAT (' ',I16,1P3D25.16)                                             SMA02900
  620 FORMAT ('1')                                                           SMA02910
      END                                                                    SMA02920
C
C***********************************************************************
C***********************************************************************
C
      SUBROUTINE INPUT                                                       INP00010
C
C     THIS SUBROUTINE READS ALL DATA
C
      IMPLICIT REAL*8(A-H,O-Z)                                               INP00020
      COMMON /NB1/ NEL,NDPT,NPEL,NDF,NEQ,NN,MM,NS,NCOUNT,NST,NSTF,NCLD,NINP00030
     1PBC,NPBC2,NBCL,NBCH,NMAT,NCON21,NDFP,NLINE,NREC7,KEL,NTLD,NBLD,NSLINP00040
     2D                                                                      INP00050
      COMMON /NB2/ NCON(20),NBC(4)                                           INP00060
      COMMON /GAUSS/ NGPS,NGPL                                               INP00070
      COMMON /B1/ ELDAT(10,3),CLOAD(4),BCON(3)                               INP00080
      COMMON /B12/ ELDATN(3),BYNP(3),STNP(3),THNP,FLOAD(3)                   INP00090
      COMMON /B2/ ELAST(6,6),COORD(3),COREL(20,3),UJNT(3),SSSS1(7),SSNN1INP00100
     1(7),ALPHA                                                              INP00110
      COMMON /CNTRL/ NLOAD                                                   INP00120
      DIMENSION TITLE(10), FMT(10), CLOAD1(3)                                INP00130
      DATA STOP/'STOP    '/                                                  INP00140
      EQUIVALENCE (CLOAD(1),CLOAD1(1)), (TITLE(1),COREL(1,1)), (FMT(1),CINP00150
     1OREL(1,2))                                                             INP00160
      NTRK(I,J) = (I-1)*NDPT+J                                               INP00170
      PI = 3.14159265359                                                     INP00180
      REWIND 9                                                               INP00190
      REWIND 14                                                              INP00200
      REWIND 15                                                              INP00210
      READ (5,290) CHKWD,NLOAD                                               INP00220
      IF (NLOAD.EQ.0) NLOAD=1                                                INP00230
      IF (CHKWD.EQ.STOP) STOP                                                INP00240
      READ (5,300) TITLE                                                     INP00250
```

```
      WRITE (6,300) TITLE                                               INP00260
C
C     READ GENERAL CHARACTERISTICS
C
      READ (5,320) NEL,NDPT,NS,NPEL,NMAT,NCLD,NPBC,NPBC2,NLINE,KEL,NTLD,INP00270
     1NBLD,NSLD,NREC7,NGPS,NGPL                                         INP00280
      WRITE (6,330) NEL,NDPT,NMAT,NS,NPBC,NPBC2,NCLD,NTLD,NBLD,NSLD,NGPSINP00290
     1,NGPL,NLOAD                                                       INP00300
      NDF = 3                                                           INP00310
      NDFP = 4                                                          INP00320
      NEQ = NDPT*NDF                                                    INP00330
      NCOUNT = NS*NS                                                    INP00340
      NST = NPEL*NDF                                                    INP00350
      NSTF = NST*NST                                                    INP00360
      NSB = 6*NST                                                       INP00370
      NN = (NEQ+NS-1)/NS                                                INP00380
      READ (5,300) TITLE                                                INP00390
      READ (5,300) FMT                                                  INP00400
      WRITE (6,310)                                                     INP00410
      WRITE (6,300) TITLE                                               INP00420
      WRITE (6,310)                                                     INP00430
      NBAND = 0                                                         INP00440
C
C     READ CONNECTIVITY
C
      IF (IABS(NMAT).EQ.1) GO TO 20                                     INP00450
      REWIND 16                                                         INP00460
      DO 10 I=1,NEL                                                     INP00470
      READ (5,FMT) NCON,NCON21                                          INP00480
      WRITE (6,FMT) NCON,NCON21                                         INP00490
      WRITE (9) NCON                                                    INP00500
      WRITE (16) NCON21                                                 INP00510
      NPELM = NPEL-1                                                    INP00520
      DO 10 J=1,NPELM                                                   INP00530
      JP = J+1                                                          INP00540
      DO 10 K=JP,NPEL                                                   INP00550
   10 NBAND = MAX0(NBAND,IABS(NCON(J)-NCON(K)))                         INP00560
      GO TO 40                                                          INP00570
   20 DO 30 I=1,NEL                                                     INP00580
      READ (5,FMT) NCON                                                 INP00590
      WRITE (6,FMT) NCON                                                INP00600
      WRITE (9) NCON                                                    INP00610
C
C     COMPUTE HALF BANDWIDTH
C
      NPELM = NPEL-1                                                    INP00620
      DO 30 J=1,NPELM                                                   INP00630
      JP = J+1                                                          INP00640
```

```
      DO 30 K=JP,NPEL                                                   INP00650
   30 NBAND = MAX0(NBAND,IABS(NCON(J)-NCON(K)))                         INP00660
   40 NBAND = (NBAND+1)*NDF                                             INP00670
      MM = (NBAND+2*(NS-1))/NS                                          INP00680
      IF (NN.LT.MM) MM = NN                                             INP00690
      WRITE (6,340) NEQ,NBAND,NN,MM,NCOUNT,NSTF,NREC7                   INP00700
C
C     CALL 'DISK' TO SET UP VARIABLES
C
      CALL DISK                                                         INP00710
      WRITE (6,310)                                                     INP00720
C
C     READ COORDINATES AND CONVERT IF NECESSARY
C
      READ (5,300) TITLE                                                INP00730
      WRITE (6,300) TITLE                                               INP00740
      WRITE (6,310)                                                     INP00750
      READ (5,300) FMT                                                  INP00760
      DO 60 I=1,NDPT                                                    INP00770
      READ (5,FMT) IEL,COORD,KND                                        INP00780
      IF (KND.EQ.0) GO TO 50                                            INP00790
      PHI = PI*COORD(2)/180.0D0                                         INP00800
      X = COORD(1)*DCOS(PHI)                                            INP00810
      Y = COORD(1)*DSIN(PHI)                                            INP00820
      COORD(1) = X                                                      INP00830
      COORD(2) = Y                                                      INP00840
   50 WRITE (6,FMT) IEL,COORD,KND                                       INP00850
   60 CALL WDISKC (I)                                                   INP00860
      WRITE (6,310)                                                     INP00870
C
C     READ MATERIAL PROPERTIES
C
      READ (5,300) TITLE                                                INP00880
      WRITE (6,300) TITLE                                               INP00890
      WRITE (6,310)                                                     INP00900
      READ (5,300) FMT                                                  INP00910
      IF (NMAT.EQ.-1) GO TO 80                                          INP00920
      DO 70 I=1,NMAT                                                    INP00930
      READ (5,FMT) N,(ELDAT(N,J),J=1,3)                                 INP00940
   70 WRITE (6,FMT) N,(ELDAT(N,J),J=1,3)                                INP00950
      WRITE (6,310)                                                     INP00960
      GO TO 110                                                         INP00970
   80 NODE2 = 1                                                         INP00980
   90 IF (NODE2.EQ.NDPT) GO TO 110                                      INP00990
      READ (5,FMT) NODE1,NODE2,YM,PR,ALPHA                              INP01000
      DO 100 I=NODE1,NODE2                                              INP01010
      ELDATN(1) = YM                                                    INP01020
      ELDATN(2) = PR                                                    INP01030
```

```
      ELDATN(3) = ALPHA
      WRITE (6,370) I,(ELDATN(J),J=1,3)
  100 CALL WDISKP (I)
      GC TO 90
C
C     READ ANY BOUNDARY CONDITIONS
C
  110 READ (5,300) TITLE
      WRITE (6,310)
      WRITE (6,300) TITLE
      WRITE (6,310)
      READ (5,300) FMT
      READ (5,FMT) NBC
      WRITE (6,FMT) NBC
      NBCL = NBC(1)
      NBC(1) = NDF*NBC(1)-NDF
      WRITE (15) NBC
      NPBCL = NPBC-1
      DO 120 I=2,NPBCL
      READ (5,FMT) NBC
      WRITE (6,FMT) NBC
      NBC(1) = NDF*NBC(1)-NDF
  120 WRITE (15) NBC
      READ (5,FMT) NBC
      WRITE (6,FMT) NBC
      NBCH = NBC(1)
      NBC(1) = NDF*NBC(1)-NDF
      WRITE (15) NBC
      IF (NPBC2.EQ.0) GO TO 140
      READ (5,300) TITLE
      WRITE (6,310)
      WRITE (6,300) TITLE
      WRITE (6,310)
      READ (5,300) FMT
      DO 130 I=1,NPBC2
      READ (5,FMT) NDBC,BCON
      WRITE (6,FMT) NDBC,BCON
  130 WRITE (22) BCON
  140 IF (NCLD.EQ.0) GO TO 160
C
C     READ ANY CONCENTRATED LOADS
C
      READ (5,300) TITLE
      WRITE (6,310)
      WRITE (6,300) TITLE
      WRITE (6,310)
      READ (5,300) FMT
      DO 150 I=1,NCLD
```

```
      READ (5,FMT) NCL,CLOAD1                                         INP01460
      WRITE (6,FMT) NCL,CLOAD1                                        INP01470
      ANCL = NCL                                                      INP01480
      CLOAD(NDFP) = ANCL                                              INP01490
  150 WRITE (14) CLOAD                                                INP01500
C
C     LOOP ON VARIOUS LOAD CONDITIONS
C
  160 DO 280 IL=1,NLOAD                                               INP01510
      IF (NTLD.EQ.0) GO TO 200                                        INP01520
      IF (NTLD.EQ.-1) GO TO 180                                       INP01530
C
C     READ TEMPERATURE DISTRIBUTIONS
C
      READ (5,300) TITLE                                              INP01540
      WRITE (6,310)                                                   INP01550
      WRITE (6,300) TITLE                                             INP01560
      WRITE (6,310)                                                   INP01570
      READ (5,300) FMT                                                INP01580
      DO 170 I=1,NDPT                                                 INP01590
      READ (5,FMT) IJT,THNP                                           INP01600
      WRITE (6,FMT) IJT,THNP                                          INP01610
      NTK = NTRK(IL,I)                                                INP01620
  170 CALL WDISKT (NTK)                                               INP01630
      GO TO 200                                                       INP01640
  180 IF (NGPS.EQ.0) GO TO 200                                        INP01650
      READ (33,350) TIME                                              INP01660
      WRITE (30) TIME                                                 INP01670
      DO 190 I=1,NDPT                                                 INP01680
      READ (33,360) IJT,THNP                                          INP01690
      NTK = NTRK(IL,I)                                                INP01700
  190 CALL WDISKT (NTK)                                               INP01710
C
C     READ ANY BODY FORCES
C
  200 IF (NBLD.EQ.0) GO TO 240                                        INP01720
      IF (NBLD.EQ.-1) GO TO 220                                       INP01730
      READ (5,300) TITLE                                              INP01740
      WRITE (6,310)                                                   INP01750
      WRITE (6,300) TITLE                                             INP01760
      WRITE (6,310)                                                   INP01770
      READ (5,300) FMT                                                INP01780
      DO 210 I=1,NDPT                                                 INP01790
      READ (5,FMT) IJT,BYNP                                           INP01800
      WRITE (6,FMT) IJT,BYNP                                          INP01810
  210 CALL WDISBF (1)                                                 INP01820
      GO TO 240                                                       INP01830
  220 IF (NGPS.EQ.0) GO TO 240                                        INP01840
```

```
      READ (34,350) TIME                                                INP01850
      WRITE (30) TIME                                                   INP01860
      DO 230 I=1,NDPT                                                   INP01870
      READ (34,370) IJT,BYNP                                            INP01880
      NTK = NTRK(IL,I)                                                  INP01890
  230 CALL WDISBF (NTK)                                                 INP01900
C
C     READ ANY DISPLACEMENTS
C
  240 IF (NSLD.EQ.0) GO TO 280                                          INP01910
      IF (NSLD.EQ.-1) GO TO 260                                         INP01920
      READ (5,300) TITLE                                                INP01930
      WRITE (6,310)                                                     INP01940
      WRITE (6,300) TITLE                                               INP01950
      WRITE (6,310)                                                     INP01960
      READ (5,300) FMT                                                  INP01970
      DO 250 I=1,NDPT                                                   INP01980
      READ (5,FMT) IJT,STNP                                             INP01990
      WRITE (6,FMT) IJT,STNP                                            INP02000
  250 CALL WDISSF (I)                                                   INP02010
      GO TO 280                                                         INP02020
  260 IF (NGPS.EQ.0) GO TO 280                                          INP02030
      READ (35,350) TIME                                                INP02040
      WRITE (30) TIME                                                   INP02050
      DO 270 I=1,NDPT                                                   INP02060
      READ (35,370) IJT,STNP                                            INP02070
      NTK = NTRK(IL,I)                                                  INP02080
  270 CALL WDISSF (NTK)                                                 INP02090
  280 CONTINUE                                                          INP02100
      RETURN                                                            INP02110
  290 FORMAT (1A8,I5)                                                   INP02120
  300 FORMAT (10A8)                                                     INP02130
  310 FORMAT (///)                                                      INP02140
  320 FORMAT (16I5)                                                     INP02150
  330 FORMAT (//,5X,'NEL....TOTAL NUMBER OF ELEMENTS......................INP02160
     1...',I3/,  5X,'NDPT...TOTAL NUMBER OF NODAL POINTS..................INP02170
     2...',I3/,  5X,'NMAT...NUMBER OF DIFFERENT MATERIALS.................INP02180
     3...',I3/,  5X,'           (MAXIMUM IS  10)',/,                        INP02190
     4           5X,'NS.....BLOCK SIZE FOR THE LARGE CAPACITY SOLVER....INP02200
     5...',I3/,  5X,'NPBC...NUMBER OF NODAL POINTS WITH BOUND. COND.....INP02210
     6...',I3/,  5X,'NPBC2..NUMBER OF NODAL POINTS WITH PREDEF. DISPL...INP02220
     7...',I3/,  5X,'NCLD...NUMBER OF NODAL POINTS WITH CONC. LOAD......INP02230
     8...',I3/,  5X,'NTLD...THERMAL LOAD PRESENT........................INP02240
     9...',I3/,  5X,'NBLD...BODY FORCE LOAD PRESENT.....................INP02250
     $...',I3/,  5X,'NSLD...INITIAL STRETCH LOAD PRESENT................INP02260
     $...',I3/,  5X,'NGPS...NUMBER OF GAUSS POINTS, STIFFNESS...........INP02270
     $...',I3/,  5X,'NGPL...NUMBER OF GAUSS POINTS, LOAD GENERATION.....INP02280
     $...',I3/,  5X,'NLOAD..NUMBER OF LOAD VECTORS......................INP02290
```

```
      $...',I3/)                                                        INP02300
  340 FORMAT (//,5X,'NEQ....TOTAL NUMBER OF EQUATIONS...................INP02310
     1...',I5/,  5X,'NBAND..HALF-BAND WIDTH OF THE SYSTEM...............INP02320
     2...',I5/,  5X,'NN.....NUMBER OF BLOCKS PER COLUMN.................INP02330
     3...',I5/,  5X,'MM.....NUMBER OF BLOCKS PER ROW....................INP02340
     4...',I5/,  5X,'NCOUNT.NUMBER OF COEFFICIENTS PER BLOCK............INP02350
     5...',I5/,  5X,'NSTF...NUMBER OF COEFFICIENTS PER ELEMENT..........INP02360
     6...',I5/,  5X,'NREC7..NUMBER OF RECORDS ON UNIT 7.................INP02370
     7...',I5/)                                                         INP02380
  350 FORMAT (F15.5)                                                    INP02390
  360 FORMAT (I5,F20.5)                                                 INP02400
  370 FORMAT (I5,1P3G20.8)                                              INP02410
      END                                                               INP02420
C
C***********************************************************************
C***********************************************************************
C
      SUBROUTINE LODGEN                                                 LDG00010
C
C     THIS SUBROUTINE GENERATES THE CONSISTENT LOAD VECTORS FOR ALL LOAD
C     CONDITIONS
C
      IMPLICIT REAL*8(A-H,O-Z)                                          LDG00020
      COMMON /NB1/ NEL,NDPT,NPEL,NDF,NEQ,NN,MM,NS,NCOUNT,NST,NSTF,NCLD,NLDG00030
     1PBC,NPBC2,NBCL,NBCH,NMAT,NCON21,NDFP,NLINE,NREC7,KEL,NTLD,NBLD,NSLLDG00040
     2D                                                                 LDG00050
      COMMON /NB2/ NCON(20),NBC(4)                                      LDG00060
      COMMON /B1/ ELDAT(10,3),CLOAD(4),BCON(3)                          LDG00070
      COMMON /B12/ ELDATN(3),BYNP(3),STNP(3),THNP,FLOAD(3)              LDG00080
      COMMON /B2/ ELAST(6,6),COORD(3),COREL(20,3),UJNT(3),SSSS1(7),SSNN1LDG00090
     1(7),ALPHA                                                         LDG00100
      COMMON /B3/ AK1(60,60),AK2(60,60),AK3(60,60),RB1(60),RB2(60),RB3(6LDG00110
     10)                                                                LDG00120
      COMMON /GAUSS/ NGPS,NGPL                                          LDG00130
      COMMON /CNTRL/ NLOAD                                              LDG00140
      DIMENSION AIA(5,5,5)                                              LDG00150
      DIMENSION XI(5,4), AI(5,4)                                        LDG00160
      DIMENSION COL(20), TEEL(20), BYEP(20,3), STEP(20,3), BTD(60,6), CELDG00170
     1LOD(60), ELDATE(20,3), B(6,60), CFT(3)                            LDG00180
      EQUIVALENCE (RB2(1),ELDATE(1)), (AK3(1,15),BYEP(1,1)), (AK3(1,17),LDG00190
     1STEP(1,1)), (AK3(1,19),BTD(1,1)), (AK3(1,1),B(1,1)), (AK3(1,33),COLDG00200
     2L(1)), (AK3(1,34),TEEL(1))                                        LDG00210
      DATA XI/0.5773502691896258,-0.5773502691896258,0.0D0,0.0D0,0.0D0,0LDG00220
     1.7745966692414 83,0.0D0,-0.7745966692414 83,0.0D0,0.0D0,0.8611363115LDG00230
     2940526,0.3399810435848563,-0.3399810435848563,-0.8611363115940526,LDG00240
     30.0D0,0.9061798459386640,0.5384693101056831,0.0D0,-0.5384693101056LDG00250
     4831,-0.9061798459386640/                                          LDG00260
      DATA AI/1.0D0,1.0D0,1.0D0,1.0D0,1.0D0,0.5555555555555556,0.888888LDG00270
```

```
     18888888888,0.5555555555555555,1.0D0,1.0D0,0.3478548451374539,0.652LDG00280
     2145154862546l,0.652145154862546l,0.3478548451374539,1.0D0,0.236926LDG00290
     388505618 91,0.4786286704993665,0.5688888888888889,0.4786286704993665,LDG00300
     40.2369268850561891/                                               LDG00310
      NTRK(I,J) = (I-1)*NDPT+J                                          LDG00320
      NGP = NGPL                                                        LDG00330
      NGPM = NGP-1                                                      LDG00340
      DO 110 I=1,NGP                                                    LDG00350
      DO 110 J=1,NGP                                                    LDG00360
      DO 110 K=1,NGP                                                    LDG00370
  110 AIA(I,J,K) = AI(I,NGPM)*AI(J,NGPM)*AI(K,NGPM)                     LDG00380
      IF (NTLD.NE.0) REWIND 24                                          LDG00390
      IF (NBLD.NE.0) REWIND 27                                          LDG00400
      IF (NSLD.NE.0) REWIND 28                                          LDG00410
      DO 340 IL=1,NLOAD                                                 LDG00420
      REWIND 9                                                          LDG00430
      REWIND 19                                                         LDG00440
      IF (NMAT.GT.1) REWIND 16                                          LDG00450
      IF (NMAT.EQ.-1) REWIND 23                                         LDG00460
      DO 120 I=1,3                                                      LDG00470
  120 FLOAD(I) = 0.0D0                                                  LDG00480
      DO 130 I=1,NDPT                                                   LDG00490
      NTRACK = I                                                        LDG00500
      NTK = NTRK(IL,NTRACK)                                             LDG00510
  130 CALL WDISLF (NTK)                                                 LDG00520
      DO 310 I=1,NEL                                                    LDG00530
      DO 140 J=1,NST                                                    LDG00540
  140 CELOD(J) = 0.0D0                                                  LDG00550
      READ (9) NCON                                                     LDG00560
      READ (19) COREL                                                   LDG00570
      IF (NTLD.NE.0) READ (24) TEEL                                     LDG00580
      IF (NBLD.NE.0) READ (27) BYEP                                     LDG00590
      IF (NSLD.NE.0) READ (28) STEP                                     LDG00600
      IF (NMAT.EQ.-1) GO TO 150                                         LDG00610
      ALPHA = ELDAT(1,3)                                                LDG00620
      IF (NMAT.EQ.+1) GO TO 160                                         LDG00630
      READ (16) NCON21                                                  LDG00640
      ALPHA = ELDAT(NCON21,3)                                           LDG00650
      CALL ELPROP (NCON21)                                              LDG00660
      GO TO 160                                                         LDG00670
  150 READ (23) ELDATE                                                  LDG00680
  160 DO 280 J=1,NGP                                                    LDG00690
      X = XI(J,NGPM)                                                    LDG00700
      DO 280 K=1,NGP                                                    LDG00710
      Y = XI(K,NGPM)                                                    LDG00720
      DO 280 L=1,NGP                                                    LDG00730
      Z = XI(L,NGPM)                                                    LDG00740
      FACT = AIA(J,K,L)                                                 LDG00750
```

```
C
C     FORM LOADS DUE TO TEMPERATURE
C
      IF (NTLD.EQ.0) GO TO 200                                          LDG00760
      IF(NMAT.EQ.-1) CALL NDPROP(X,Y,Z,2)                               LDG00761
      CALL FORMK (X,Y,Z,DTJ,2)                                          LDG00770
      DO 170 II=1,NST                                                   LDG00780
      DO 170 JJ=1,6                                                     LDG00790
      BTD(II,JJ) = 0.0D0                                                LDG00800
      DO 170 KK=1,6                                                     LDG00810
  170 BTD(II,JJ) = BTD(II,JJ)+B(KK,II)*ELAST(KK,JJ)                     LDG00820
      CALL SHAPE (X,Y,Z,COL)                                            LDG00830
      CFT(1) = 0.0D0                                                    LDG00840
      DO 180 II=1,NPEL                                                  LDG00850
  180 CFT(1) = CFT(1)+COL(II)*TEEL(II)                                  LDG00860
      CFT(1) = CFT(1)*ALPHA*DTJ                                         LDG00870
      DO 190 II=1,NST                                                   LDG00880
  190 CELOD(II) = CELOD(II)+(BTD(II,1)+BTD(II,2)+BTD(II,3))*CFT(1)*FACT LDG00890
C
C     FORM LOADS DUE TO BODY FORCES
C
  200 IF (NBLD.EQ.0) GO TO 230                                          LDG00900
      CALL SHAPE (X,Y,Z,COL)                                            LDG00910
      CALL FORMK (X,Y,Z,DTJ,1)                                          LDG00920
      CFT(1) = 0.0D0                                                    LDG00930
      CFT(2) = 0.0D0                                                    LDG00940
      CFT(3) = 0.0D0                                                    LDG00950
      DO 210 II=1,NPEL                                                  LDG00960
      DO 210 JJ=1,3                                                     LDG00970
      CFT(JJ) = CFT(JJ)+COL(II)*BYEP(II,JJ)                             LDG00980
  210 CONTINUE                                                          LDG00990
      DO 220 II=1,NPEL                                                  LDG01000
      DO 220 KK=1,NDF                                                   LDG01010
      IK = 3*II+KK-3                                                    LDG01020
  220 CELOD(IK) = CELOD(IK)+COL(II)*CFT(KK)*DTJ*FACT                    LDG01030
C
C     FORM LOADS DUE TO DISPLACEMENT
C
  230 IF (NSLD.EQ.0) GO TO 280                                          LDG01040
      CALL FORMK (X,Y,Z,DTJ,2)                                          LDG01050
      IF(NMAT.EQ.-1) CALL NDPROP(X,Y,Z,0)                               LDG01051
      DO 240 II=1,NST                                                   LDG01060
      DO 240 JJ=1,6                                                     LDG01070
      BTD(II,JJ) = 0.0D0                                                LDG01080
      DO 240 KK=1,6                                                     LDG01090
  240 BTD(II,JJ) = BTD(II,JJ)+B(KK,II)*ELAST(KK,JJ)                     LDG01100
      DO 250 JJ=1,NST                                                   LDG01110
      DC 250 II=1,NST                                                   LDG01120
```

```
  250 AK1(II,JJ) = 0.0D0
      DO 260 II=1,NST
      DO 260 JJ=1,NST
      DO 260 KK=1,6
  260 AK1(II,JJ) = AK1(II,JJ)+BTD(II,KK)*B(KK,JJ)
      DO 270 II=1,NST
      DO 270 JJ=1,NPEL
      DO 270 KK=1,NDF
      IJ = 3*JJ+KK-3
  270 CELOD(II) = CELOD(II)+AK1(II,IJ)*STEP(JJ,KK)*DTJ*FACT
  280 CONTINUE
      DO 300 II=1,NPEL
      NTRACK = NCON(II)
      NTK = NTRK(IL,NTRACK)
      CALL RDISLF (NTK)
      DO 290 JJ=1,NDF
      IJ = 3*II+JJ-3
  290 FLOAD(JJ) = CELOD(IJ)+FLOAD(JJ)
  300 CALL WDISLF (NTK)
  310 CONTINUE
C
C     APPLY ANY BOUNDARY CONDITIONS
C
      ZRO = 0.0D0
      REWIND 15
      READ (15) NBC
      NBC1 = (NBC(1)+NDF)/NDF
      DO 330 I=1,NDPT
      NTK = NTRK(IL,I)
      CALL RDISLF (NTK)
      IF (I.NE.NBC1) GO TO 330
      DO 320 J=1,3
      IF (NBC(J+1).NE.0) FLOAD(J)=ZRO
  320 CONTINUE
C
C     STORE COMPLETED LOAD VECTORS
C
      CALL WDISLF (NTK)
      IF (I.EQ.NDPT) GO TO 330
      READ (15) NBC
      NBC1 = (NBC(1)+NDF)/NDF
  330 CONTINUE
  340 CONTINUE
      RETURN
      END
C
C***********************************************************************
C***********************************************************************
```

```
C
      SUBROUTINE BCOND                                                  BCD00010
C
C     THIS SUBROUTINE APPLIES ANY BOUNDARY CONDITIONS TO APPROPRIATE
C     BLOCKS OF THE TOTAL STIFFNESS MATRIX
C
      IMPLICIT REAL*8(A-H,O-Z)                                          BCD00020
      COMMON /NB1/ NEL,NDPT,NPEL,NDF,NEQ,NN,MM,NS,NCOUNT,NST,NSTF,NCLD,NBCD00030
     1PBC,NPBC2,NBCL,NBCH,NMAT,NCON21,NDFP,NLINE,NREC7,KEL,NTLD,NBLD,NSLBCD00040
     2D                                                                 BCD00050
      COMMON /NB2/ NCON(20),NBC(4)                                      BCD00060
      COMMON /B1/ ELDAT(10,3),CLOAD(4),BCON(3)                          BCD00070
      COMMON /B3/ AK1(60,60),AK2(60,60),AK3(60,60),RB1(60),RB2(60),RB3(6BCD00080
     10)                                                                BCD00090
      COMMON /CNTRL/ NLOAD                                              BCD00100
      DIMENSION AKW1(3600), AKW2(3600)                                  BCD00110
      EQUIVALENCE (AKW1(1),AK1(1,1)), (AKW2(1),AK2(1,1))                BCD00120
      NTRK(I,J) = (I-1)*(MM)+J                                          BCD00130
      NTKR(I,J) = (I-1)*NN+J                                            BCD00140
      ABIGN = 1.0D20                                                    BCD00150
      REWIND 15                                                         BCD00160
      READ (15) NBC                                                     BCD00170
      NBC1 = NBC(1)+NDF                                                 BCD00180
      INDEX = 1                                                         BCD00190
      DO 50 I=1,NN                                                      BCD00200
      IH = I*NS                                                         BCD00210
      IL = (I-1)*NS+1                                                   BCD00220
      IF (NBC1.GT.IH) GO TO 50                                          BCD00230
      NTK = NTRK(I,1)                                                   BCD00240
      CALL RDDISK (NTK,AKW1)                                            BCD00250
   10 IBL = NBC(1)+1-NS*(I-1)                                           BCD00260
      IBH = IBL+NDF-1                                                   BCD00270
      IC = 1                                                            BCD00280
      DO 20 K=IBL,IBH                                                   BCD00290
      IC = IC+1                                                         BCD00300
      IF (NBC(IC).NE.1) GO TO 20                                        BCD00310
      AK1(K,K) = AK1(K,K)+ABIGN                                         BCD00320
   20 CONTINUE                                                          BCD00330
      IF (INDEX.EQ.NPBC) GO TO 30                                       BCD00340
      READ (15) NBC                                                     BCD00350
      NBC1 = NBC(1)+NDF                                                 BCD00360
      INDEX = INDEX+1                                                   BCD00370
      IF (NBC1.LE.IH) GO TO 10                                          BCD00380
      GO TO 40                                                          BCD00390
   30 CALL WRDISK (NTK,AKW1)                                            BCD00400
      RETURN                                                            BCD00410
   40 CALL WRDISK (NTK,AKW1)                                            BCD00420
   50 CONTINUE                                                          BCD00430
```

```
      RETURN
C
C     ENTRY 'BCOND2' IS FOR ANY PREDEFINED DISPLACEMENTS
C
      ENTRY BCOND2
      DIMENSION BCON1(60,2)
      EQUIVALENCE (BCON1(1,1),AK3(1,1))
      ABIGN = 1.0D20
      DC 310 IL=1,NLOAD
      I1 = 1
      I5 = 0
      MMF = NN-MM+1
      IFR = (NBCL*NDF+NS-1)/NS
      IR1 = 1
      IF (IFR.LE.MM) GO TO 60
      IR1 = IFR-MM+1
   60 DO 290 I=IR1,NN
      IFR1 = IFR-I+1
      IF (IFR1.GT.MM) GO TO 290
      REWIND 15
      REWIND 22
      NTKR1 = NTKR(IL,I)
      CALL RDSKR1 (NTKR1)
      DO 70 L=1,I1
   70 READ (15) NBC
      NBC1 = NBC(1)+NDF
      NBC2 = NBC1/NDF
      IF (I5.EQ.0) GO TO 90
      DO 80 L=1,I5
   80 READ (22) BCON
   90 IFR1 = IFR
      J3 = 2
      IF (IFR.NE.I) GO TO 200
      I4 = 0
      IH = I*NS
      NTK1 = NTRK(I,1)
      CALL RDDISK (NTK1,AKW1)
      I61 = 0
  100 I6 = 0
      DO 150 K=1,NDF
      K1 = K+1
      M = NBC(K1)+1
      II = NBC(1)-(I-1)*NS+K
      GO TO (150,140,110), M
  110 IF (I6.NE.0) GO TO 120
      READ (22) BCON
      I5 = I5+1
      I6 = 1
```

```
      I61 = 1                                                          BCD00890
  120 DO 130 L=1,NS                                                    BCD00900
  130 RB1(L) = RB1(L)-AK1(L,II)*BCON(K)                                BCD00910
      I4 = I4+1                                                        BCD00920
      BCON1(I4,1) = II                                                 BCD00930
      BCON1(I4,2) = BCON(K)                                            BCD00940
  140 AK1(II,II) = AK1(II,II)+ABIGN                                    BCD00950
  150 CONTINUE                                                         BCD00960
      J3 = MM+1                                                        BCD00970
      MML = I/MMF-(I-MMF)/MMF                                          BCD00980
      MM1 = MM-(I-MMF)*MML                                             BCD00990
      J4 = MM1                                                         BCD01000
      IF (NBC2.EQ.NBCH) GO TO 160                                      BCD01010
      READ (15) NBC                                                    BCD01020
      NBC1 = NBC(1)+NDF                                                BCD01030
      NBC2 = NBC1/NDF                                                  BCD01040
      I1 = I1+1                                                        BCD01050
      IF (NBC1.LE.IH) GO TO 100                                        BCD01060
      J1 = (NBC1+NS-1)/NS                                              BCD01070
      J2 = I+1                                                         BCD01080
      J3 = J1-I+1                                                      BCD01090
      IFR = J1                                                         BCD01100
      IF (I61.EQ.0) GO TO 190                                          BCD01110
      IH = J1*NS                                                       BCD01120
      IF (J1.EQ.J2) GO TO 190                                          BCD01130
      J4 = J3-1                                                        BCD01140
      IF (J4.GT.MM) J4 = MM                                            BCD01150
  160 IF (I.EQ.NN) GO TO 300                                           BCD01160
      DO 180 J=2,J4                                                    BCD01170
      NTK2 = NTRK(I,J)                                                 BCD01180
      CALL RDDISK (NTK2,AKW2)                                          BCD01190
      NTKR2 = J+I-1                                                    BCD01200
      NTK = NTKR(IL,NTKR2)                                             BCD01210
      CALL RDSKR2 (NTK)                                                BCD01220
      DO 170 L1=1,I4                                                   BCD01230
      IR = BCON1(L1,1)                                                 BCD01240
      DO 170 L2=1,NS                                                   BCD01250
  170 RB2(L2) = RB2(L2)-AK2(IR,L2)*BCON1(L1,2)                         BCD01260
  180 CALL WDSKR2 (NTK)                                                BCD01270
  190 CALL WRDISK (NTK1,AKW1)                                          BCD01280
  200 IF (J3.GT.MM) GO TO 280                                          BCD01290
      MML = I/MMF-(I-MMF)/MMF                                          BCD01300
      MM1 = MM-(I-MMF)*MML                                             BCD01310
      DC 270 J=J3,MM1                                                  BCD01320
      IH = (J+(I-1))*NS                                                BCD01330
      NTK2 = NTRK(I,J)                                                 BCD01340
      CALL RDDISK (NTK2,AKW2)                                          BCD01350
      IHL = (J+I-2)*NS+1                                               BCD01360
```

```
      IPD = (IH+1-NBC1)*(IHL-NBC1)                                      BCD01370
      IF (IPD.GT.0) GO TO 250                                           BCD01380
  210 I6 = 0                                                            BCD01390
      DO 240 K=1,NDF                                                    BCD01400
      K1 = K+1                                                          BCD01410
      IF (NBC(K1).NE.2) GO TO 240                                       BCD01420
      IF (I6.NE.0) GO TO 220                                            BCD01430
      READ (22) BCON                                                    BCD01440
      I6 = 1                                                            BCD01450
  220 II = NBC(1)+K-(I+J-2)*NS                                          BCD01460
      DO 230 L=1,NS                                                     BCD01470
  230 RB1(L) = RB1(L)-AK2(L,II)*BCON(K)                                 BCD01480
  240 CONTINUE                                                          BCD01490
      IF (NBC2.EQ.NBCH) GO TO 250                                       BCD01500
      READ (15) NBC                                                     BCD01510
      NBC1 = NBC(1)+NDF                                                 BCD01520
      NBC2 = NBC1/NDF                                                   BCD01530
      IF (NBC1.GT.IH) GO TO 250                                         BCD01540
      GO TO 210                                                         BCD01550
  250 IFR6 = IFR1*I61                                                   BCD01560
      IF (IFR6.NE.1) GO TO 270                                          BCD01570
      NTKR2 = J+I-1                                                     BCD01580
      NTK = NTKR(IL,NTKR2)                                              BCD01590
      CALL RDSKR2 (NTK)                                                 BCD01600
      DO 260 L1=1,I4                                                    BCD01610
      IR = BCON1(L1,1)                                                  BCD01620
      DO 260 L2=1,NS                                                    BCD01630
  260 RB2(L2) = RB2(L2)-AK2(IR,L2)*BCON1(L1,2)                          BCD01640
      CALL WDSKR2 (NTK)                                                 BCD01650
  270 CONTINUE                                                          BCD01660
  280 CALL WDSKR1 (NTKR1)                                               BCD01670
  290 CONTINUE                                                          BCD01680
      RETURN                                                            BCD01690
  300 CALL WDSKR1 (NTKR1)                                               BCD01700
      CALL WRDISK (NTK1,AKW1)                                           BCD01710
  310 CONTINUE                                                          BCD01720
      RETURN                                                            BCD01730
      END                                                               BCD01740
C
C***********************************************************************
C***********************************************************************
C
      SUBROUTINE SOLVE                                                  SLV00010
C
C     THIS SUBROUTINE IS THE BLOCK SOLVER PATTERNED AFTER A ROUTINE
C     ORIGINALLY WRITTEN BY PROF. J.G. CANTIN AT THE U.S. NAVAL
C     POSTGRADUATE SCHOOL
C
```

```
      IMPLICIT REAL*8(A-H,O-Z)                                          SLV00020
      COMMON /NB1/ NEL,NDPT,NPEL,NDF,NEQ,NN,MM,NS,NCOUNT,NST,NSTF,NCLD,NSLV00030
     1PBC,NPBC2,NBCL,NBCH,NMAT,NCON21,NDFP,NLINE,NREC7,KEL,NTLD,NBLD,NSLSLV00040
     2D                                                                 SLV00050
      COMMON /B3/ AT1(60,60),AT2(60,60),AT3(60,60),RT1(60),RT2(60),RT3(6SLV00060
     10)                                                                SLV00070
      COMMON /CNTRL/ NLOAD                                              SLV00080
      DIMENSION AK1(3600), AK2(3600), AK3(3600), RB1(60), RB2(60), RB3(6SLV00090
     10)                                                                SLV00100
      EQUIVALENCE (AK1(1),AT1(1,1)), (AK2(1),AT2(1,1)), (AK3(1),AT3(1,1)SLV00110
     1), (RB1(1),RT1(1)), (RB2(1),RT2(1)), (RB3(1),RT3(1))              SLV00120
      NTRK(I,J) = (I-1)*(MM)+J                                          SLV00130
      NTKR(I,J) = (I-1)*NN+J                                            SLV00140
C
C     ROW REDUCTION
C
      KMM = MM                                                          SLV00150
      N = 0                                                             SLV00160
   10 N = N+1                                                           SLV00170
      NTK = NTRK(N,1)                                                   SLV00180
      CALL RDDISK (NTK,AK2)                                             SLV00190
      CALL SYMINV (AK2,NS,RB1,RB2,IFLG)                                 SLV00200
      IF (IFLG.EQ.1) WRITE (6,150) N                                    SLV00210
      IF (IFLG.EQ.1) STOP                                               SLV00220
      IF (IFLG.EQ.2) WRITE (6,160) N                                    SLV00230
      NTR = N                                                           SLV00240
      DO 20 IL=1,NLOAD                                                  SLV00250
      NTK = NTKR(IL,N)                                                  SLV00260
      CALL RDSKR1 (NTK)                                                 SLV00270
      CALL MULT (AK2,RB1,RB2,NS,NS,1)                                   SLV00280
      CALL WDSKR2 (NTK)                                                 SLV00290
   20 CONTINUE                                                          SLV00300
      IF (N.EQ.NN) GO TO 100                                            SLV00310
      IF (N.GT.(NN-MM+1)) KMM=KMM-1                                     SLV00320
      DO 40 K=2,KMM                                                     SLV00330
      NTK = NTRK(N,K)                                                   SLV00340
      CALL RDDISK (NTK,AK1)                                             SLV00350
      CALL MULT (AK2,AK1,AK3,NS,NS,NS)                                  SLV00360
      CALL WRDISK (NTK,AK3)                                             SLV00370
      DO 30 I=1,NS                                                      SLV00380
      II = (I-1)*NS                                                     SLV00390
      DO 30 J=1,NS                                                      SLV00400
      IL = II+J                                                         SLV00410
      IR = I+(J-1)*NS                                                   SLV00420
   30 AK3(IL) = AK1(IR)                                                 SLV00430
      NTK = NTRK(NN,K)                                                  SLV00440
      CALL WRDISK (NTK,AK3)                                             SLV00450
   40 CONTINUE                                                          SLV00460
```

```
      DO 90 L=2,KMM
      I = N+L-1
      IF (I.GT.NN) GO TO 90
      J = 0
      NTK = NTRK(NN,L)
      CALL RDDISK (NTK,AK2)
      DO 60 K=L,KMM
      J = J+1
      NTK = NTRK(N,K)
      CALL RDDISK (NTK,AK1)
      CALL MULT (AK2,AK1,AK3,NS,NS,NS)
      NTK = NTRK(I,J)
      CALL RDDISK (NTK,AK1)
      DO 50 I1=1,NCOUNT
   50 AK1(I1) = AK1(I1)-AK3(I1)
      CALL WRDISK (NTK,AK1)
   60 CONTINUE
      DO 80 IL=1,NLOAD
      NTK = NTKR(IL,N)
      CALL RDSKR2 (NTK)
      CALL MULT (AK2,RB2,RB3,NS,NS,1)
      NTR = I
      NTK = NTKR(IL,NTR)
      CALL RDSKR1 (NTK)
      DO 70 I1=1,NS
   70 RB1(I1) = RB1(I1)-RB3(I1)
   80 CALL WDSKR1 (NTK)
   90 CONTINUE
      GO TO 10
C
C     BACK SUBSTITUTION
C
  100 N = N-1
      IF (N.EQ.0) GO TO 140
      NT1 = N
      DO 130 IL=1,NLOAD
      NTK = NTKR(IL,NT1)
      CALL RDSKR3 (NTK)
      DO 120 K=2,KMM
      L = N+K-1
      IF (L.GT.NN) GO TO 120
      NTK = NTRK(N,K)
      CALL RDDISK (NTK,AK1)
      NTR = L
      NTK1 = NTKR(IL,NTR)
      CALL RDSKR1 (NTK1)
      CALL MULT (AK1,RB1,RB2,NS,NS,1)
      DO 110 K1=1,NS
```

```
  110 RB3(K1) = RB3(K1)-RB2(K1)                                         SLV00920
  120 CONTINUE                                                          SLV00930
      NTK = NTKR(IL,NT1)                                                SLV00940
      CALL WDSKR3 (NTK)                                                 SLV00950
  130 CONTINUE                                                          SLV00960
      KMM = KMM+1                                                       SLV00970
      IF (KMM.GT.MM) KMM = MM                                           SLV00980
      GO TO 100                                                         SLV00990
  140 RETURN                                                            SLV01000
  150 FORMAT ('0',10X,'***** BLOCK NO.',I3,' IS SINGULAR. EXECUTION STOPSLV01010
     1S *****')                                                         SLV01020
  160 FORMAT ('0',10X,'***** BLOCK NO.',I3,' IS NEARLY SINGULAR. EXECUTISLV01030
     1ON CONTINUES *****')                                              SLV01040
      END                                                               SLV01050
C
C***********************************************************************
C***********************************************************************
C
      SUBROUTINE DISP (ILL)                                             DSP00010
C
C     THIS SUBROUTINE FORMS NODAL DISPLACEMENTS AND REACTIONS AT
C     BOUNDARY CONDITION NODES
C
      IMPLICIT REAL*8(A-H,O-Z)                                          DSP00020
      COMMON /NB1/ NEL,NDPT,NPEL,NDF,NEQ,NN,MM,NS,NCOUNT,NST,NSTF,NCLD,NDSP00030
     1PBC,NPBC2,NBCL,NBCH,NMAT,NCON21,NDFP,NLINE,NREC7,KEL,NTLD,NBLD,NSLDSP00040
     2D                                                                 DSP00050
      COMMON /NB2/ NCON(20),NBC(4)                                      DSP00060
      COMMON /B3/ AK1(60,60),AK2(60,60),AK3(60,60),RB1(60),RB2(60),RB3(6DSP00070
     10)                                                                DSP00080
      COMMON /CNTRL/ NLOAD                                              DSP00090
      DIMENSION REACT(4), REACT1(3)                                     DSP00100
      EQUIVALENCE (REACT(1),RB2(1)), (REACT(2),REACT1(1))               DSP00110
      NTKR(I,J) = (I-1)*NN+J                                            DSP00120
      ABIGN = 1.0D20                                                    DSP00130
      ZRO = 0.0D0                                                       DSP00140
      REWIND 15                                                         DSP00150
      REWIND 17                                                         DSP00160
      READ (15) NBC                                                     DSP00170
      NBC1 = NBC(1)+NDF                                                 DSP00180
      INDEX = 1                                                         DSP00190
      NPBL = NS/NDF                                                     DSP00200
      JLL = NDPT-(NDPT/NPBL)*NPBL                                       DSP00210
      IF (JLL.EQ.0) JLL = NPBL                                          DSP00220
      NL1 = 0                                                           DSP00230
      WRITE (6,250)                                                     DSP00240
      DO 220 I=1,NN                                                     DSP00250
      NTK = NTKR(ILL,I)                                                 DSP00260
```

```
      CALL RDSKR1 (NTK)                                                 DSP00270
      CALL WDISKB (I)                                                   DSP00280
      IM1 = I-1                                                         DSP00290
      ICTI = IM1*NPBL                                                   DSP00300
      IH = I*NS                                                         DSP00310
      IL = IM1*NS+1                                                     DSP00320
  110 ICHECK = (NBC1-IH)*(IL-NBC1)                                      DSP00330
      IF (ICHECK.LT.0) GO TO 160                                        DSP00340
      DO 120 J=1,NDFP                                                   DSP00350
  120 REACT(J) = ZRO                                                    DSP00360
      I1 = NBC(1)-NS*(I-1)+1                                            DSP00370
      I2 = I1+1                                                         DSP00380
      I3 = I2+1                                                         DSP00390
      IF (NBC(2).EQ.0) GO TO 130                                        DSP00400
      REACT(2) = RB1(I1)*ABIGN                                          DSP00410
      RB1(I1) = ZRO                                                     DSP00420
  130 IF (NBC(3).EQ.0) GO TO 140                                        DSP00430
      REACT(3) = RB1(I2)*ABIGN                                          DSP00440
      RB1(I2) = ZRO                                                     DSP00450
  140 IF (NBC(4).EQ.0) GO TO 150                                        DSP00460
      REACT(4) = RB1(I3)*ABIGN                                          DSP00470
      RB1(I3) = ZRO                                                     DSP00480
  150 ANBC = NBC(1)/NDF+1                                               DSP00490
      REACT(1) = ANBC                                                   DSP00500
      WRITE (17) REACT                                                  DSP00510
      IF (INDEX.EQ.NPBC) GO TO 160                                      DSP00520
      READ (15) NBC                                                     DSP00530
      NBC1 = NBC(1)+NDF                                                 DSP00540
      INDEX = INDEX+1                                                   DSP00550
      GO TO 110                                                         DSP00560
C
C     PRINT DISPLACEMENTS
C
  160 NL2 = NL1+NPBL                                                    DSP00570
      JL = NPBL                                                         DSP00580
      IF (I.EQ.NN) JL = JLL                                             DSP00590
      IF (NL2.GT.NLINE) GO TO 180                                       DSP00600
      DO 170 M=1,JL                                                     DSP00610
      ICT = ICTI+M                                                      DSP00620
      K1 = (M-1)*NDF+1                                                  DSP00630
      K2 = K1+2                                                         DSP00640
  170 WRITE (6,270) ICT,(RB1(N1),N1=K1,K2)                              DSP00650
      NL1 = NL2                                                         DSP00660
      GO TO 220                                                         DSP00670
  180 J2 = NLINE-NL1                                                    DSP00680
      IF (J2.LE.0) GO TO 200                                            DSP00690
      DO 190 N=1,J2                                                     DSP00700
      ICT = ICTI+N                                                      DSP00710
```

```
      K1 = (N-1)*NDF+1                                                  DSP00720
      K2 = K1+2                                                         DSP00730
  190 WRITE (6,270) ICT,(RB1(N2),N2=K1,K2)                              DSP00740
  200 NL1 = NPBL-J2                                                     DSP00750
      IF (JL.LE.0) GO TO 220                                            DSP00760
      WRITE (6,250)                                                     DSP00770
      J1 = J2+1                                                         DSP00780
      DO 210 M1=J1,JL                                                   DSP00790
      ICT = ICT1+M1                                                     DSP00800
      K1 = (M1-1)*NDF+1                                                 DSP00810
      K2 = K1+2                                                         DSP00820
  210 WRITE (6,270) ICT,(RB1(N3),N3=K1,K2)                              DSP00830
  220 CONTINUE                                                          DSP00840
C
C     PRINT REACTIONS
C     CHECK EQUILIBRIUM
C
      SUMX = ZRO                                                        DSP00850
      SUMY = ZRO                                                        DSP00860
      SUMZ = ZRO                                                        DSP00870
      I2 = ZRO                                                          DSP00880
      REWIND 17                                                         DSP00890
      I1 = 1                                                            DSP00900
  230 I2 = I2+NLINE                                                     DSP00910
      IF (NPBC.LE.I2) I2 = NPBC                                         DSP00920
      WRITE (6,260)                                                     DSP00930
      DO 240 L3=I1,I2                                                   DSP00940
      READ (17) REACT                                                   DSP00950
      ICT = REACT(1)                                                    DSP00960
      WRITE (6,270) ICT,REACT1                                          DSP00970
      SUMX = SUMX+REACT(2)                                              DSP00980
      SUMY = SUMY+REACT(3)                                              DSP00990
      SUMZ = SUMZ+REACT(4)                                              DSP01000
  240 CONTINUE                                                          DSP01010
      I1 = I2+1                                                         DSP01020
      IF (I2.LT.NPBC) GO TO 230                                         DSP01030
      WRITE (6,280) SUMX,SUMY,SUMZ                                      DSP01040
      RETURN                                                            DSP01050
  250 FORMAT (1H1,//,5X,'  D-I-S-P-L-A-C-E-M-E-N-T-S   ',               DSP01060
     1                //,5X,5H N.PT,12X,3H  X,23X,2H Y,23X,2H Z,//)     DSP01070
  260 FORMAT (1H1,//,5X,'  R-E-A-C-T-I-O-N-S  ',                        DSP01080
     1                //,5X,5H N.PT,12X,3H  X,23X,2H Y,23X,2H Z,//)     DSP01090
  270 FORMAT (5X,I5,3(1PD25.16))                                        DSP01100
  280 FORMAT (///,' EQUILIBRIUM CHECK ',/,10X,3(1PD25.16))              DSP01110
      END                                                               DSP01120
C
C***********************************************************************
C***********************************************************************
```

```
C
      SUBROUTINE PRNSTR (STR,S1,S2,S3)                                  PST00010
C
C     THIS SUBROUTINE DETERMINES PRINCIPAL STRESS VALUES BY CUBIC
C     EQUATION FORMULA
C
      IMPLICIT REAL*8(A-H,O-Z)                                          PST00020
      COMPLEX *16AC,BC,CC                                               PST00030
      DIMENSION STR(1)                                                  PST00040
      DATA CC/(0.0D0,1.732050808D0)/,CR/0.333333333333D0/               PST00050
      SX = STR(1)                                                       PST00060
      SY = STR(2)                                                       PST00070
      SZ = STR(3)                                                       PST00080
      TXY = STR(4)                                                      PST00090
      TYZ = STR(5)                                                      PST00100
      TXZ = STR(6)                                                      PST00110
      PHI = -(SX+SY+SZ)                                                 PST00120
      PHI1 = PHI                                                        PST00130
      THETA = SX*SY+SY*SZ+SX*SZ-(TXY*TXY+TXZ*TXZ+TXY*TXY)               PST00140
      PSI = -(SX*SY*SZ+2.0*TYZ*TXZ*TXY-(SX*TYZ*TYZ+SY*TXZ*TXZ+SZ*TXY*TXYPST00150
     1))                                                                PST00160
      A = THETA-PHI*PHI/3.0D0                                           PST00170
      B = 2.0D0*PHI*PHI*PHI/27.0D0-PHI*THETA/3.0D0+PSI                  PST00180
      AB = B*B/4.0D0+A*A*A/27.0D0                                       PST00190
      IF (AB.LT.0.0D0) GO TO 150                                        PST00200
      AB = DSQRT(AB)                                                    PST00210
      BPAB = -B/2.0D0+AB                                                PST00220
      BMAB = -B/2.0D0-AB                                                PST00230
      IF (BPAB.LT.0.0D0) GO TO 110                                      PST00240
      AC = BPAB**CR                                                     PST00250
      GO TO 120                                                         PST00260
  110 AC = -((-BPAB)**CR)                                               PST00270
  120 IF (BMAB.LT.0.0D0) GO TO 130                                      PST00280
      BC = BMAB**CR                                                     PST00290
      GO TO 140                                                         PST00300
  130 BC = -((-BMAB)**CR)                                               PST00310
  140 S1 = DREAL(AC+BC)                                                 PST00320
      S2 = DREAL(-(AC+BC)/2.0D0+CC*(A-B)/2.0D0)                         PST00330
      S3 = DREAL(-(AC+BC)/2.0D0-CC*(A-B)/2.0D0)                         PST00340
      S1 = S1-PHI1/3.0D0                                                PST00350
      S2 = S2-PHI1/3.0D0                                                PST00360
      S3 = S3-PHI1/3.0D0                                                PST00370
      RETURN                                                            PST00380
  150 AB = (-B/2.0D0)/DSQRT(-A*A*A/27.0D0)                              PST00390
      PHI = DARCOS(AB)                                                  PST00400
      FACT = 2.0D0*DSQRT(-A/3.0D0)                                      PST00410
      PHI3 = PHI/3.0D0                                                  PST00420
      S1 = FACT*DCOS(PHI3)                                              PST00430
```

```
      S2 = FACT*DCOS(PHI3+2.094395103D0)                                PST00440
      S3 = FACT*DCOS(PHI3+4.18879206D0)                                 PST00450
      S1 = S1-PHII/3.0D0                                                PST00460
      S2 = S2-PHII/3.0D0                                                PST00470
      S3 = S3-PHII/3.0D0                                                PST00480
      RETURN                                                            PST00490
      END                                                               PST00500
C
C***********************************************************************
C***********************************************************************
C
      SUBROUTINE ELPRUP (I)                                             ELP00010
C
C     THIS SUBROUTINE FORMS THE ELASTIC MATRIX FOR ELEMENTS
C
      IMPLICIT REAL*8(A-H,O-Z)                                          ELP00020
      COMMON /B1/ ELDAT(10,3),CLOAD(4),BCON(3)                          ELP00030
      COMMON /B2/ ELAST(6,6),COORD(3),COREL(20,3),UJNT(3),SSSS1(7),SSNN1ELP00040
     1(7),ALPHA                                                         ELP00050
      DO 110 L=1,6                                                      ELP00060
      DO 110 J=1,6                                                      ELP00070
  110 ELAST(L,J) = 0.0D0                                                ELP00080
      E = ELDAT(I,1)                                                    ELP00090
      PR = ELDAT(I,2)                                                   ELP00100
      ER = E/((1.0D0+PR)*(1.0D0-2.0D0*PR))                              ELP00110
      EK = ER*(1.0D0-PR)                                                ELP00120
      EL = ER*PR                                                        ELP00130
      ELAST(1,1) = EK                                                   ELP00140
      ELAST(1,2) = EL                                                   ELP00150
      ELAST(1,3) = EL                                                   ELP00160
      ELAST(2,1) = EL                                                   ELP00170
      ELAST(2,2) = EK                                                   ELP00180
      ELAST(2,3) = EL                                                   ELP00190
      ELAST(3,1) = EL                                                   ELP00200
      ELAST(3,2) = EL                                                   ELP00210
      ELAST(3,3) = EK                                                   ELP00220
      ELAST(4,4) = ER*(1.0D0-2.0D0*PR)/2.0D0                            ELP00230
      ELAST(5,5) = ELAST(4,4)                                           ELP00240
      ELAST(6,6) = ELAST(4,4)                                           ELP00250
      RETURN                                                            ELP00260
      END                                                               ELP00270
C
C***********************************************************************
C***********************************************************************
C
      SUBROUTINE FORMK (X,Y,Z,DTJ,INDIC)                                FMK00010
C
C     THIS SUBROUTINE CALCULATES SHAPE FUNCTION DERIVATIVE VALUES,
```

```
C     FORMS THE JACOBIAN AND ITS DETERMINANT
C
      IMPLICIT REAL*8(A-H,O-Z)                                          FMK00020
      COMMON /NB1/ NEL,NDPT,NPEL,NDF,NEQ,NN,MM,NS,NCOUNT,NST,NSTF,NCLD,NFMK00030
     1PBC,NPBC2,NDCL,NBCH,NMAT,NCON21,NDFP,NLINE,NREC7,KEL,NTLD,NBLD,NSLFMK00040
     2D                                                                 FMK00050
      COMMON /B2/ ELAST(6,6),COORD(3),COREL(20,3),UJNT(3),SSSS1(7),SSNN1FMK00060
     1(7),ALPHA                                                         FMK00070
      COMMON /B3/ AK1(60,60),AK2(60,60),AK3(60,60),RB1(60),RB2(60),RB3(6FMK00080
     10)                                                                FMK00090
      DIMENSION STK(60,60), AK(60,60), B(6,60), B1(60,6), W1(3,20), DNX(FMK00100
     13,20), AJIN(3,3), AJ(3,3), CORDG(20,3)                            FMK00110
      EQUIVALENCE (STK(1,1),AK1(1,1)), (AK(1,1),AK2(1,1)), (B(1,1),AK3(1FMK00120
     1,1)), (B1(1,1),AK3(1,7)), (W1(1,1),RB3(1))                        FMK00130
      DATA CORDG/1.0D0,0.0D0,-1.0D0,-1.0D0,-1.0D0,0.0D0,1.0D0,1.0D0,1.0DFMK00140
     10,-1.0D0,-1.0D0,1.0D0,1.0D0,0.0D0,-1.0D0,-1.0D0,-1.0D0,0.0D0,1.0D0FMK00150
     2,1.0D0,1.0D0,1.0D0,1.0D0,0.0D0,-1.0D0,-1.0D0,-1.0D0,0.0D0,1.0D0,1.FMK00160
     30D0,-1.0D0,-1.0D0,1.0D0,1.0D0,1.0D0,0.0D0,-1.0D0,-1.0D0,-1.0D0,0.0FMK00170
     4D0,1.0D0,1.0D0,1.0D0,1.0D0,1.0D0,1.0D0,1.0D0,1.0D0,0.0D0,0.0D0,0.0FMK00180
     5D0,0.0D0,-1.0D0,-1.0D0,-1.0D0,-1.0D0,-1.0D0,-1.0D0,-1.0D0,-1.0D0/ FMK00190
      DF(C,D,E,X1,Y1,Z1) = X1*(1.0D0+D*Y1)*(1.0D0+E*Z1)*(2.0D0*C*X1+D*Y1FMK00200
     1+E*Z1-1.0D0)/8.0D0                                                FMK00210
      D2(C,D,E,Y1,Z1) = -C*(1.0D0+D*Y1)*(1.0D0+E*Z1)/2.0D0              FMK00220
      D4(C,E,Y1,Z1) = (1.0D0-C*C)*Y1*(1.0D0+E*Z1)/4.0D0                 FMK00230
C
C     CORNER NODES
C
      DO 110 I=1,2                                                      FMK00240
      II = 12*(I-1)+1                                                   FMK00250
      IT = II+6                                                         FMK00260
      DO 110 J=II,IT,2                                                  FMK00270
      X1 = CORDG(J,1)                                                   FMK00280
      Y1 = CORDG(J,2)                                                   FMK00290
      Z1 = CORDG(J,3)                                                   FMK00300
      W1(1,J) = DF(X,Y,Z,X1,Y1,Z1)                                      FMK00310
      W1(2,J) = DF(Y,Z,X,Y1,Z1,X1)                                      FMK00320
  110 W1(3,J) = DF(Z,X,Y,Z1,X1,Y1)                                      FMK00330
      DO 120 K=1,2                                                      FMK00340
      II = (K-1)*12+2                                                   FMK00350
      IT = II+4                                                         FMK00360
      DO 120 L=II,IT,4                                                  FMK00370
      X1 = CORDG(L,1)                                                   FMK00380
      Y1 = CORDG(L,2)                                                   FMK00390
      Z1 = CORDG(L,3)                                                   FMK00400
      W1(1,L) = D2(X,Y,Z,Y1,Z1)                                         FMK00410
      W1(2,L) = D4(X,Z,Y1,Z1)                                           FMK00420
  120 W1(3,L) = D4(X,Y,Z1,Y1)                                           FMK00430
C
```

```
C
C     MIDSIDE NODES
C
      DO 130 K=1,2                                                     FMK00440
      II = (K-1)*12+4                                                  FMK00450
      IT = II+4                                                        FMK00460
      DO 130 L=II,IT,4                                                 FMK00470
      X1 = CORDG(L,1)                                                  FMK00480
      Y1 = CORDG(L,2)                                                  FMK00490
      Z1 = CORDG(L,3)                                                  FMK00500
      W1(1,L) = D4(Y,Z,X1,Z1)                                          FMK00510
      W1(2,L) = D2(Y,Z,X,Z1,X1)                                        FMK00520
  130 W1(3,L) = D4(Y,X,Z1,X1)                                          FMK00530
      DO 140 L=9,12                                                    FMK00540
      X1 = CORDG(L,1)                                                  FMK00550
      Y1 = CORDG(L,2)                                                  FMK00560
      Z1 = CORDG(L,3)                                                  FMK00570
      W1(1,L) = D4(Z,Y,X1,Y1)                                          FMK00580
      W1(2,L) = D4(Z,X,Y1,X1)                                          FMK00590
  140 W1(3,L) = D2(Z,X,Y,X1,Y1)                                        FMK00600
      DO 150 I=1,3                                                     FMK00610
      DO 150 J=1,3                                                     FMK00620
      AJ(I,J) = 0.0D0                                                  FMK00630
      DO 150 K=1,NPEL                                                  FMK00640
  150 AJ(I,J) = AJ(I,J)+W1(I,K)*COREL(K,J)                             FMK00650
C
C     FORM JACOBIAN
C
      DTJ = AJ(1,1)*AJ(2,2)*AJ(3,3)+AJ(1,2)*AJ(2,3)*AJ(3,1)+AJ(1,3)*AJ(2FMK00660
     1,1)*AJ(3,2)-AJ(3,1)*AJ(2,2)*AJ(1,3)-AJ(3,2)*AJ(2,3)*AJ(1,1)-AJ(3,3FMK00670
     2)*AJ(2,1)*AJ(1,2)                                                FMK00680
      IF (INCIC.EQ.1) RETURN                                           FMK00690
      AJIN(1,1) = (AJ(2,2)*AJ(3,3)-AJ(3,2)*AJ(2,3))/DTJ                FMK00700
      AJIN(2,1) = -(AJ(2,1)*AJ(3,3)-AJ(3,1)*AJ(2,3))/DTJ               FMK00710
      AJIN(3,1) = (AJ(2,1)*AJ(3,2)-AJ(3,1)*AJ(2,2))/DTJ                FMK00720
      AJIN(1,2) = -(AJ(1,2)*AJ(3,3)-AJ(3,2)*AJ(1,3))/DTJ               FMK00730
      AJIN(2,2) = (AJ(1,1)*AJ(3,3)-AJ(3,1)*AJ(1,3))/DTJ                FMK00740
      AJIN(3,2) = -(AJ(1,1)*AJ(3,2)-AJ(3,1)*AJ(1,2))/DTJ               FMK00750
      AJIN(1,3) = (AJ(1,2)*AJ(2,3)-AJ(2,2)*AJ(1,3))/DTJ                FMK00760
      AJIN(2,3) = -(AJ(1,1)*AJ(2,3)-AJ(2,1)*AJ(1,3))/DTJ               FMK00770
      AJIN(3,3) = (AJ(1,1)*AJ(2,2)-AJ(2,1)*AJ(1,2))/DTJ                FMK00780
      DO 160 I=1,3                                                     FMK00790
      DO 160 J=1,NPEL                                                  FMK00800
      DNX(I,J) = 0.0D0                                                 FMK00810
      DO 160 K=1,3                                                     FMK00820
  160 DNX(I,J) = DNX(I,J)+AJIN(I,K)*W1(K,J)                            FMK00830
      DO 170 I=1,6                                                     FMK00840
      DO 170 J=1,NST                                                   FMK00850
      B1(J,I) = 0.0D0                                                  FMK00860
```

```
  170 B(I,J) = 0.0D0                                                    FMK00870
C
C     FORM MATRIX OF SHAPE FUNCTION DERIVATIVES
C
      DO 180 I=1,NPEL                                                   FMK00880
      I1 = 3*I-2                                                        FMK00890
      I2 = I1+1                                                         FMK00900
      I3 = I2+1                                                         FMK00910
      B(1,I1) = DNX(1,I)                                                FMK00920
      B(2,I2) = DNX(2,I)                                                FMK00930
      B(3,I3) = DNX(3,I)                                                FMK00940
      B(4,I1) = DNX(2,I)                                                FMK00950
      B(4,I2) = DNX(1,I)                                                FMK00960
      B(5,I2) = DNX(3,I)                                                FMK00970
      B(5,I3) = DNX(2,I)                                                FMK00980
      B(6,I3) = DNX(1,I)                                                FMK00990
  180 B(6,I1) = DNX(3,I)                                                FMK01000
      IF (INDIC.EQ.2) RETURN                                            FMK01010
C
C     FORM INTEGRAND OF STIFFNESS MATRIX FORMULA
C
      DO 200 I=1,NST                                                    FMK01020
      DO 190 J=1,3                                                      FMK01030
      DO 190 K=1,3                                                      FMK01040
  190 B1(I,J) = B1(I,J)+B(K,I)*ELAST(K,J)                               FMK01050
      DO 200 L=4,6                                                      FMK01060
  200 B1(I,L) = B1(I,L)+B(L,I)*ELAST(L,L)                               FMK01070
      DO 210 I=1,NST                                                    FMK01080
      DO 210 J=I,NST                                                    FMK01090
      AK(I,J) = 0.0D0                                                   FMK01100
      DO 210 K=1,6                                                      FMK01110
  210 AK(I,J) = AK(I,J)+B1(I,K)*B(K,J)                                  FMK01120
      DO 220 I=1,NST                                                    FMK01130
      DO 220 J=I,NST                                                    FMK01140
      AK(I,J) = AK(I,J)*DTJ                                             FMK01150
  220 AK(J,I) = AK(I,J)                                                 FMK01160
      RETURN                                                            FMK01170
      END                                                               FMK01180
C
C***********************************************************************
C***********************************************************************
C
      SUBROUTINE FSTF                                                   FST00010
C
C     THIS SUBROUTINE FORMS THE STIFFNESS MATRICES
C
      IMPLICIT REAL*8(A-H,O-Z)                                          FST00020
      COMMON /NB1/ NEL,NDPT,NPEL,NDF,NEQ,NN,MM,NS,NCOUNT,NST,NSTF,NCLD,NFST00030
```

```
     1PBC,NPBC2,NBCL,NBCH,NMAT,NCON21,NDFP,NLINE,NREC7,KEL,NTLD,NBLD,NSLFST00040
     2D                                                                FST00050
      COMMON /B2/ ELAST(6,6),COORD(3),COREL(20,3),UJNT(3),SSSS1(7),SSNN1FST00060
     1(7),ALPHA                                                         FST00070
      COMMON /B3/ AK1(60,60),AK2(60,60),AK3(60,60),RB1(60),RB2(60),RB3(6FST00080
     10)                                                                FST00090
      DIMENSION STK(3600), AK(3600), B(360), ELDATN(20,3)               FST00100
      EQUIVALENCE (AK1(1,1),STK(1)), (AK2(1,1),AK(1)), (AK3(1,1),B(1)), FST00110
     1(RB2(1),ELDATN(1))                                                FST00120
      REWIND 19                                                         FST00130
      IF (NMAT.EQ.-1) GO TO 150                                         FST00140
      IF (NMAT.EQ.1) GO TO 130                                          FST00150
C
C     FOR USE WITH MORE THAN ONE ELEMENT PROPERTY
C
      REWIND 16                                                         FST00160
      N2 = -1                                                           FST00170
      DO 120 I=1,NEL                                                    FST00180
      READ (19) COREL                                                   FST00190
      READ (16) NCON21                                                  FST00200
      N = NCON21                                                        FST00210
      IF (N.EQ.N2) GO TO 110                                            FST00220
      CALL ELPROP (N)                                                   FST00230
      N2 = N                                                            FST00240
  110 CALL CUB                                                          FST00250
      CALL WDISK1 (I)                                                   FST00260
  120 CONTINUE                                                          FST00270
      RETURN                                                            FST00280
C
C     ONE ELEMENT PROPERTY
C
  130 CALL ELPROP (1)                                                   FST00290
      DO 140 I=1,NEL                                                    FST00300
      READ (19) COREL                                                   FST00310
      CALL CUB                                                          FST00320
      CALL WDISK1 (I)                                                   FST00330
  140 CONTINUE                                                          FST00340
      RETURN                                                            FST00350
C
C     NODAL PROPERTIES
C
  150 REWIND 23                                                         FST00360
      DO 160 I=1,NEL                                                    FST00370
      READ (19) COREL                                                   FST00380
      READ (23) ELDATN                                                  FST00390
      CALL CUB                                                          FST00400
      CALL WDISK1 (I)                                                   FST00410
  160 CONTINUE                                                          FST00420
```

```
      RETURN
      END
C
C*****************************************************************************
C*****************************************************************************
C
      SUBROUTINE MERGE
C
C     THIS SUBROUTINE FORMS THE TOTAL STIFFNESS MATRIX
C
      IMPLICIT REAL*8(A-H,O-Z)
      COMMON /NB1/ NEL,NDPT,NPEL,NDF,NEQ,NN,MM,NS,NCOUNT,NST,NSTF,NCLD,N
     1PBC,NPBC2,NBCL,NBCH,NMAT,NCON21,NDFP,NLINE,NREC7,KEL,NTLD,NBLD,NSL
     2D
      COMMON /NB2/ NCON(20),NBC(4)
      COMMON /B3/ AK1(60,60),AK2(60,60),AK3(60,60),RB1(60),RB2(60),RB3(6
     10)
      DIMENSION A(60,60), S(60,60), B(3600), T(3600), LM(20)
      EQUIVALENCE (AK2(1,1),A(1,1),B(1)), (AK1(1,1),S(1,1),T(1)), (LM(1)
     1,RB1(1))
      NTRK(I,J) = (I-1)*(MM)+J
      ZRO = 0.0D0
      DO 190 I=1,NN
      DO 190 J=1,MM
      DO 110 I1=1,NS
      DO 110 J1=1,NS
  110 A(I1,J1) = ZRO
      REWIND 9
      DO 180 K=1,NEL
      READ (9) NCON
      DO 120 L=1,NPEL
  120 LM(L) = NDF*NCON(L)-NDF
      LMIN = 5000
      LMAX = 0
      DO 130 I2=1,NPEL
      LMIN = MIN0(LMIN,LM(I2))
  130 LMAX = MAX0(LMAX,LM(I2))
      LMIN = LMIN+1
      LMAX = LMAX+1
      NSI = NS*I
      NSIM = NSI-NS
      IF (LMIN.GT.NSI) GO TO 180
      IF (LMAX.LE.NSIM) GO TO 180
      CALL RDISK1 (K)
      DO 170 II=1,NPEL
      DO 160 JJ=1,NPEL
      DO 150 KK=1,NDF
      III = LM(II)+KK-NS*(I-1)
```

```
      IF ((III.GT.NS).OR.(III.LT.1)) GO TO 150                          MRG00400
      KKK = NDF*II-NDF+KK                                               MRG00410
      DO 140 LL=1,NDF                                                   MRG00420
      JJJ = LM(JJ)+LL-NS*(J+I-2)                                        MRG00430
      IF ((JJJ.GT.NS).OR.(JJJ.LT.1)) GO TO 140                          MRG00440
      LLL = NDF*JJ-NDF+LL                                               MRG00450
      A(III,JJJ) = A(III,JJJ)+S(KKK,LLL)                                MRG00460
  140 CONTINUE                                                          MRG00470
  150 CONTINUE                                                          MRG00480
  160 CONTINUE                                                          MRG00490
  170 CONTINUE                                                          MRG00500
  180 CONTINUE                                                          MRG00510
C
C     WRITE COMPLETED BLOCK ON DISK
C
      NTK = NTRK(I,J)                                                   MRG00520
      CALL WRDISK (NTK,B)                                               MRG00530
  190 CONTINUE                                                          MRG00540
      IF (NN*NS.EQ.NEQ) GO TO 210                                       MRG00550
      NTK = NTRK(NN,1)                                                  MRG00560
      CALL RDDISK (NTK,B)                                               MRG00570
      DO 200 I=1,NS                                                     MRG00580
      IF (DABS(A(I,I)).LT.1.0D-14) A(I,I)=1.0D0                         MRG00590
  200 CONTINUE                                                          MRG00600
      CALL WRDISK (NTK,B)                                               MRG00610
  210 CONTINUE                                                          MRG00620
      RETURN                                                            MRG00630
      END                                                               MRG00640
C
C***********************************************************************
C***********************************************************************
C
      SUBROUTINE STRESS (I,ILL)                                         STR00010
C
C     THIS SUBROUTINE COMPUTES NODAL STRESSES
C
      IMPLICIT REAL*8(A-H,O-Z)                                          STR00020
      COMMON /NB1/ NEL,NDPT,NPEL,NDF,NEQ,NN,MM,NS,NCOUNT,NST,NSTF,NCLD,NSTR00030
     1PBC,NPBC2,NBCL,NBCH,NMAT,NCON21,NDFP,NLINE,NREC7,KEL,NTLD,NBLD,NSLSTR00040
     2D                                                                 STR00050
      COMMON /NB2/ NCON(20),NBC(4)                                      STR00060
      COMMON /B1/ ELDAT(10,3),CLOAD(4),BCON(3)                          STR00070
      COMMON /B2/ ELAST(6,6),COORD(3),COREL(20,3),UJNT(3),SSSS1(7),SSNN1STR00080
     1(7),ALPHA                                                         STR00090
      COMMON /B3/ AK1(60,60),AK2(60,60),AK3(60,60),RB1(60),RB2(60),RB3(6STR00100
     10)                                                                STR00110
      DIMENSION SNELM(20,6), UEL(60), SSELM(20,6), SSNN(6), B(6,60), SSSSTR00120
     1S(6), CORDG(20,3), LM(20), TEEL(20), ELDATE(20,3)                 STR00130
```

```
      EQUIVALENCE (AK3(1,1),B(1)), (AK3(1,13),SNELM(1)), (AK3(1,15),SSELSTR00140
     1M(1)), (AK3(1,17),SSNN(1)), (AK3(7,17),SSSS(1)), (UEL(1),RB1(1)), STR00150
     2(AK3(1,34),TEEL(1)), (RB2(1),ELDATE(1,1))                        STR00160
      DATA CORDG/1.0D0,0.0D0,-1.0D0,-1.0D0,-1.0D0,0.0D0,1.0D0,1.0D0,1.0DSTR00170
     10,-1.0D0,-1.0D0,1.0D0,1.0D0,0.0D0,-1.0D0,-1.0D0,-1.0D0,0.0D0,1.0D0STR00180
     2,1.0D0,1.0D0,1.0D0,1.0D0,0.0D0,-1.0D0,-1.0D0,-1.0D0,0.0D0,1.0D0,1.STR00190
     30D0,-1.0D0,-1.0D0,1.0D0,1.0D0,1.0D0,0.0D0,-1.0D0,-1.0D0,-1.0D0,0.0STR00200
     4D0,1.0D0,1.0D0,1.0D0,1.0D0,1.0D0,1.0D0,1.0D0,1.0D0,0.0D0,0.0D0,0.0STR00210
     5D0,0.0D0,-1.0D0,-1.0D0,-1.0D0,-1.0D0,-1.0D0,-1.0D0,-1.0D0,-1.0D0/ STR00220
      NTRK(I,J) = (I-1)*NDPT+J                                          STR00230
      READ (9) NCON                                                     STR00240
      IF (KEL.EQ.0) GO TO 110                                           STR00250
      WRITE (6,220) I                                                   STR00260
  110 DO 130 J=1,NPEL                                                   STR00270
      LM(J) = NCON(J)                                                   STR00280
      JJ = LM(J)                                                        STR00290
      JEL = 3*(J-1)                                                     STR00300
      CALL RDISK3 (JJ)                                                  STR00310
      DO 120 K=1,NDF                                                    STR00320
      J2 = JEL+K                                                        STR00330
  120 UEL(J2) = UJNT(K)                                                 STR00340
  130 CONTINUE                                                          STR00350
C
C     COMPUTE ELEMENT STRAINS
C
      DO 190 J=1,NPEL                                                   STR00360
      JJ = NCON(J)                                                      STR00370
      X = CORDG(J,1)                                                    STR00380
      Y = CORDG(J,2)                                                    STR00390
      Z = CORDG(J,3)                                                    STR00400
      IF(NMAT.EQ.-1) CALL NDPROP(X,Y,Z,2)                               STR00401
      CALL FORMK (X,Y,Z,DTJ,2)                                          STR00410
      DO 140 J1=1,6                                                     STR00420
      SSSS(J1) = 0.0D0                                                  STR00430
      SSNN(J1) = 0.0D0                                                  STR00440
      DO 140 J2=1,NST                                                   STR00450
  140 SSNN(J1) = SSNN(J1)+B(J1,J2)*UEL(J2)                              STR00460
      CORT = 0.0D0                                                      STR00470
      IF (NTLD.EQ.0) GO TO 150                                          STR00480
      CORT = ALPHA*TEEL(J)                                              STR00490
C
C     COMPUTE NODAL STRESSES AND STRAINS
C
  150 DO 160 J1=1,3                                                     STR00500
      DO 160 J2=1,3                                                     STR00510
  160 SSSS(J1) = SSSS(J1)+(SSNN(J2)-CORT)*ELAST(J1,J2)                  STR00520
      DO 170 K1=4,6                                                     STR00530
  170 SSSS(K1) = SSSS(K1)+SSNN(K1)*ELAST(K1,K1)                         STR00540
```

```
C
C     FORMS STRESS AND STRAIN VECTORS
C
      NTK = NTRK(ILL,JJ)                                                STR00550
      CALL RDISKS (NTK)                                                 STR00560
      CALL RDISKN (JJ)                                                  STR00570
      SSSS1(7) = SSSS1(7)+1.0D0                                         STR00580
      SSNN1(7) = SSNN1(7)+1.0D0                                         STR00590
      DO 180 K=1,6                                                      STR00600
      SNELM(J,K) = SSNN(K)                                              STR00610
      SSELM(J,K) = SSSS(K)                                              STR00620
      SSSS1(K) = SSSS1(K)+SSSS(K)                                       STR00630
      SSNN1(K) = SSNN1(K)+SSNN(K)                                       STR00640
  180 CONTINUE                                                          STR00650
      CALL WDISKS (NTK)                                                 STR00660
      CALL WDISKN (JJ)                                                  STR00670
  190 CONTINUE                                                          STR00680
      IF (KEL.EQ.0) GO TO 210                                           STR00690
C
C     PRINT ELEMENT NODAL VALUES IF REQUIRED
C
      DO 200 K=1,NPEL                                                   STR00700
      WRITE (6,230) LM(K),(SNELM(K,L),L=1,6)                            STR00710
      WRITE (6,230) LM(K),(SSELM(K,M),M=1,6)                            STR00720
  200 WRITE (6,240)                                                     STR00730
  210 CONTINUE                                                          STR00740
      RETURN                                                            STR00750
  220 FORMAT (1H1,//' S-T-R-A-I-N-S / S-T-R-E-S-S-E-S FOR ELEMENT ',I4, STR00760
     1 //,2X,'JOINT',6X,                                                STR00770
     2 'SNX/SSX',8X,'SNY/SSY',8X,'SNZ/SSZ',8X,'SNXY/SSXY',6X,'SNYZ/SSYZ'STR00780
     3,6X,'SNZX/SSZX',//)                                               STR00790
  230 FORMAT (2X,I4,4X,6(1PD15.6))                                      STR00800
  240 FORMAT (1H )                                                      STR00810
      END                                                               STR00820
C
C**********************************************************************
C**********************************************************************
C
      SUBROUTINE CUB                                                    CUB00010
C
C     THIS SUBROUTINE PERFORMS CUBIC INTEGRATION
C
      IMPLICIT REAL*8(A-H,O-Z)                                          CUB00020
      COMMON /NB1/ NEL,NDPT,NPEL,NDF,NEQ,NN,MM,NS,NCOUNT,N,NSTF,NCLD,NPBCUB00030
     1C,NPBC2,NBCL,NBCH,NMAT,NCON21,NDFP,NLINE,NREC7,KEL,NTLD,NBLD,NSLD CUB00040
      COMMON /B3/ AK1(60,60),AK2(60,60),AK3(60,60),RB1(60),RB2(60),RB3(6CUB00050
     10)                                                                CUB00060
      COMMON /GAUSS/ NGPS,NGPL                                          CUB00070
```

```
      DIMENSION STK(60,60), AK(60,60), B(6,60)                          CUB00080
      DIMENSION AIA(5,5,5)                                              CUB00090
      DIMENSION XI(5,4), AI(5,4)                                        CUB00100
      EQUIVALENCE (STK(1,1),AK1(1,1)), (AK(1,1),AK2(1,1)), (B(1,1),AK3(1CUB00110
     1,1))                                                              CUB00120
      DATA XI/0.5773502691896258,-0.5773502691896258,0.0D0,0.0D0,0.0D0,0CUB00130
     1.774596669241483,0.0D0,-0.774596669241483,0.0D0,0.0D0,0.8611363115CUB00140
     2940526,0.3399810435848563,-0.3399810435848563,-0.8611363115940526,CUB00150
     30.0D0,0.9061798459386640,0.5384693101056831,0.0D0,-0.5384693101056CUB00160
     4831,-0.9061798459386640/                                          CUB00170
      DATA AI/1.0D0,1.0D0,1.0D0,1.0D0,1.0D0,0.5555555555555556,0.888888CUB00180
     18888888888,0.5555555555555555,1.0D0,1.0D0,0.3478548451374539,0.652CUB00190
     21451548625461,0.6521451548625461,0.3478548451374539,1.0D0,0.236926CUB00200
     38850561891,0.4786286704993665,0.5688888888888889,0.4786286704993665,CUB00210
     40.2369268850561891/                                               CUB00220
      NGP = NGPS                                                        CUB00230
      NGPM = NGP-1                                                      CUB00240
      DO 110 I=1,NGP                                                    CUB00250
      DO 110 J=1,NGP                                                    CUB00260
      DO 110 K=1,NGP                                                    CUB00270
  110 AIA(I,J,K) = AI(I,NGPM)*AI(J,NGPM)*AI(K,NGPM)                     CUB00280
      DO 120 I=1,N                                                      CUB00290
      DO 120 J=1,N                                                      CUB00300
  120 STK(I,J) = 0.0D0                                                  CUB00310
      DO 130 I=1,NGP                                                    CUB00320
      X = XI(I,NGPM)                                                    CUB00330
      DO 130 J=1,NGP                                                    CUB00340
      Y = XI(J,NGPM)                                                    CUB00350
      DO 130 K=1,NGP                                                    CUB00360
      Z = XI(K,NGPM)                                                    CUB00370
      FACT = AIA(I,J,K)                                                 CUB00380
      IF(NMAT.EQ.-1) CALL NDPROP(X,Y,Z,0)                               CUB00381
      CALL FORMK (X,Y,Z,DTJ,0)                                          CUB00390
      DO 130 L=1,N                                                      CUB00400
      DO 130 M=1,N                                                      CUB00410
  130 STK(L,M) = STK(L,M)+FACT*AK(L,M)                                  CUB00420
      DO 140 I=1,N                                                      CUB00430
      DO 140 J=I,N                                                      CUB00440
      STK(I,J) = (STK(I,J)+STK(J,I))*0.5D0                              CUB00450
  140 STK(J,I) = STK(I,J)                                               CUB00460
      RETURN                                                            CUB00470
      END                                                               CUB00480
C
C***********************************************************************
C***********************************************************************
C
      SUBROUTINE SHAPE (XI,ETA,ZETA,COL)                                SHP00010
C
```

```
C     THIS SUBROUTINE COMPUTES SHAPE FUNCTION VALUES
C
      IMPLICIT REAL*8(A-H,O-Z)                                          SHP00020
      DIMENSION CORDX(4), CORDY(4), CORDZ(4), COL(1)                    SHP00030
      DATA CORDX/1.0D0,-1.0D0,-1.0D0,1.0D0/                             SHP00040
      DATA CORDY/1.0D0,1.0D0,-1.0D0,-1.0D0/                             SHP00050
      DATA CORDZ/1.0D0,-1.0D0,1.0D0,-1.0D0/                             SHP00060
      FQC(XI,ETA,ZETA,X1,Y1,Z1)=(1.0D0+XI*X1)*(1.0D0+ETA*Y1)*(1.0D0+ZETASHP00070
     1*Z1)*(XI*X1+ETA*Y1+ZETA*Z1-2.0D0)/8.0D0                           SHP00080
      FQM(XI,ETA,ZETA,Y1,Z1)=(1.0D0-XI*XI)*(1.0D0+ETA*Y1)*(1.0D0+ZETA*Z1SHP00090
     1)/4.0D0                                                           SHP00100
      DO 110 IZ=1,2                                                     SHP00110
      II = 12*(IZ-1)+1                                                  SHP00120
      IT = II+6                                                         SHP00130
      IX = 0                                                            SHP00140
      DO 110 ICOL=II,IT,2                                               SHP00150
      IX = IX+1                                                         SHP00160
      Y1 = CORDY(IX)                                                    SHP00170
      X1 = CORDX(IX)                                                    SHP00180
      Z1 = CORDZ(IZ)                                                    SHP00190
  110 COL(ICOL) = FQC(XI,ETA,ZETA,X1,Y1,Z1)                             SHP00200
      IC9 = 8                                                           SHP00210
      DO 120 IZ=1,2                                                     SHP00220
      II = 12*(IZ-1)+2                                                  SHP00230
      IT = II+4                                                         SHP00240
      DO 120 IC2=II,IT,4                                                SHP00250
      IC4 = IC2+2                                                       SHP00260
      IC9 = IC9+1                                                       SHP00270
      IN = IC9-8                                                        SHP00280
      X1 = CORDX(IN)                                                    SHP00290
      Y1 = CORDY(IN)                                                    SHP00300
      Z1 = CORDZ(IN)                                                    SHP00310
      COL(IC2) = FQM(XI,ETA,ZETA,Z1,Y1)                                 SHP00320
      COL(IC4) = FQM(ETA,ZETA,XI,Y1,-Z1)                                SHP00330
  120 COL(IC9) = FQM(ZETA,XI,ETA,X1,Y1)                                 SHP00340
      RETURN                                                            SHP00350
      END                                                               SHP00360
C
C***********************************************************************
C***********************************************************************
C
      SUBROUTINE MULT (A,B,C,NRA,NCA,NCB)                               MUL00010
C
C     THIS SUBROUTINE FORMS C=A*B
C
      IMPLICIT REAL*8(A-H,O-Z)                                          MUL00020
      DIMENSION A(1), B(1), C(1)                                        MUL00030
      ZRO = 0.0D0                                                       MUL00040
```

```
      DO 110  I=1,NRA                                                   MUL00050
      DO 110  J=1,NCB                                                   MUL00060
      IC =  I+(J-1)*NRA                                                 MUL00070
      C(IC) =  ZRO                                                      MUL00080
      DO 110  K=1,NCA                                                   MUL00090
      IA =  I+(K-1)*NRA                                                 MUL00100
      IB =  K+(J-1)*NCA                                                 MUL00110
      C(IC) =  C(IC)+A(IA)*B(IB)                                        MUL00120
  110 CCNTINUE                                                          MUL00130
      RETURN                                                            MUL00140
      END                                                               MUL00150
C
C***********************************************************************
C***********************************************************************
C
      SUBROUTINE DISK                                                   DSK00010
C
C     THIS SUBROUTINE CCNTRCLS ALL READ AND WRITE OPERATIONS TO THE
C     VARIOUS FILES USED IN THE EXTERNAL STORAGE.
C
      IMPLICIT REAL*8(A-H,O-Z)                                          DSK00020
      COMMON /NB1/NEL,NDPT,NPEL,NDF,NEQ,NN,MM,NS,NCOUNT,NST,NSTF,NCLD,  DSK00030
     1NPBC,NPBC2,NBCL,NBCH,NMAT,NCON21,NDFP,NLINE,NREC7,KEL,NTLD,NBLD,  DSK00040
     2NSLD                                                              DSK00050
      CCMMON /B12/ ELDATN(3),BYNP(3),STNP(3),THNP,FLOAD(3)              DSK00060
      COMMON /B2/ ELAST(6,6),COORD(3),COREL(20,3),UJNT(3),SSSS1(7),     DSK00070
     1SSNN1(7),ALPHA                                                    DSK00080
      CCMMON /B3/ AK1(60,60),AK2(60,60),AK3(60,60),RB1(60),RB2(60),     DSK00090
     1RB3(60)                                                           DSK00100
      CCMMON /IFILE/ I7,I8,I10,I11,I12,I13,I18,I20,I21,I25,I26,I29      DSK00110
      DIMENSION STK1(900),STK2(900),STK3(900),STK4(900),A(1)            DSK00120
      EQUIVALENCE (STK1(1),AK1(1,1)),(STK2(1),AK1(1,16)),               DSK00130
     1(STK3(1),AK1(1,31)),(STK4(1),AK1(1,46))                           DSK00140
      NCCUNT=NS*NS                                                      DSK00150
      LREC7=NCOUNT/NREC7                                                DSK00160
      NS1=NS/NDF                                                        DSK00170
      RETURN                                                            DSK00180
      ENTRY WRDISK(NTRACK,A)                                            DSK00190
      I7=(NTRACK-1)*NREC7+1                                             DSK00200
      N1=1                                                              DSK00210
      DC 10 L=1,NREC7                                                   DSK00220
      N2=L*LREC7                                                        DSK00230
      WRITE(7'I7)            (A(J),J=N1,N2)                             DSK00240
   10 N1=N2+1                                                           DSK00250
      RETURN                                                            DSK00260
      ENTRY RDDISK(NTRACK,A)                                            DSK00270
      I7=(NTRACK-1)*NREC7+1                                             DSK00280
      N1=1                                                              DSK00290
```

```
      DO 20 L=1,NREC7                                                   DSK00300
      N2=L*LREC7                                                        DSK00310
      READ(7'I7,ERR=500)            (A(J),J=N1,N2)                      DSK00320
   20 N1=N2+1                                                           DSK00330
      RETURN                                                            DSK00340
      ENTRY   WDISK1(NTRACK)                                            DSK00350
      I8=(NTRACK-1)*4+1                                                 DSK00360
      WRITE(8'I8)    STK1                                               DSK00370
      WRITE(8'I8)    STK2                                               DSK00380
      WRITE(8'I8)    STK3                                               DSK00390
      WRITE(8'I8)    STK4                                               DSK00400
      RETURN                                                            DSK00410
      ENTRY   RDISK1(NTRACK,A)                                          DSK00420
      I8=(NTRACK-1)*4+1                                                 DSK00430
      READ(8'I8,ERR=500)    STK1                                        DSK00440
      READ(8'I8,ERR=500)    STK2                                        DSK00450
      READ(8'I8,ERR=500)    STK3                                        DSK00460
      READ(8'I8,ERR=500)    STK4                                        DSK00470
      RETURN                                                            DSK00480
      ENTRY WDISKC(NTRACK)                                              DSK00490
      WRITE(10'NTRACK)    COORD                                         DSK00500
      RETURN                                                            DSK00510
      ENTRY RDISKC(NTRACK)                                              DSK00520
      READ(10'NTRACK,ERR=500)       COORD                               DSK00530
      RETURN                                                            DSK00540
      ENTRY WDISKP(NTRACK)                                              DSK00550
      WRITE(11'NTRACK) ELDATN                                           DSK00560
      RETURN                                                            DSK00570
      ENTRY RDISKP(NTRACK)                                              DSK00580
      READ(11'NTRACK,ERR=500) ELDATN                                    DSK00590
      RETURN                                                            DSK00600
      ENTRY RDISKS(NTRACK)                                              DSK00610
      READ(12'NTRACK,ERR=500) SSSS1                                     DSK00620
      RETURN                                                            DSK00630
      ENTRY WDISKS(NTRACK)                                              DSK00640
      WRITE(12'NTRACK) SSSS1                                            DSK00650
      RETURN                                                            DSK00660
      ENTRY RDISKN(NTRACK)                                              DSK00670
      READ(13'NTRACK,ERR=500) SSNN1                                     DSK00680
      RETURN                                                            DSK00690
      ENTRY WDISKN(NTRACK)                                              DSK00700
      WRITE(13'NTRACK) SSNN1                                            DSK00710
      RETURN                                                            DSK00720
      ENTRY WDISKT(NTRACK)                                              DSK00730
      WRITE(18'NTRACK) THNP                                             DSK00740
      RETURN                                                            DSK00750
      ENTRY RDISKT(NTRACK)                                              DSK00760
      READ(18'NTRACK,ERR=500) THNP                                      DSK00770
```

```
      RETURN                                                            DSK00780
      ENTRY RDSKR1(NTRACK)                                              DSK00790
      READ(20'NTRACK,ERR=500)    RB1                                    DSK00800
      RETURN                                                            DSK00810
      ENTRY RDSKR2(NTRACK)                                              DSK00820
      READ(20'NTRACK,ERR=500)    RB2                                    DSK00830
      RETURN                                                            DSK00840
      ENTRY RDSKR3(NTRACK)                                              DSK00850
      READ(20'NTRACK,ERR=500)    RB3                                    DSK00860
      RETURN                                                            DSK00870
      ENTRY WDSKR1(NTRACK)                                              DSK00880
      WRITE(20'NTRACK)  RB1                                             DSK00890
      RETURN                                                            DSK00900
      ENTRY WDSKR2(NTRACK)                                              DSK00910
      WRITE(20'NTRACK)  RB2                                             DSK00920
      RETURN                                                            DSK00930
      ENTRY WDSKR3(NTRACK)                                              DSK00940
      WRITE(20'NTRACK)  RB3                                             DSK00950
      RETURN                                                            DSK00960
      ENTRY WDISKB(NTRACK)                                              DSK00970
      I21=(NTRACK-1)*NS1+1                                              DSK00980
      J1=1                                                              DSK00990
      J2=NDF                                                            DSK01000
      DO 600  I=1,NS1                                                   DSK01010
      WRITE(21'I21)      (RB1(J),J=J1,J2)                               DSK01020
      J1=J2+1                                                           DSK01030
      J2=J2+NDF                                                         DSK01040
  600 CONTINUE                                                          DSK01050
      RETURN                                                            DSK01060
      ENTRY RDISKB(NTRACK)                                              DSK01070
      READ (21'NTRACK,ERR=500)       UJNT                               DSK01080
      RETURN                                                            DSK01090
      ENTRY WDISBF(NTRACK)                                              DSK01100
      WRITE(25'NTRACK) BYNP                                             DSK01110
      RETURN                                                            DSK01120
      ENTRY RDISBF(NTRACK)                                              DSK01130
      READ(25'NTRACK,ERR=500) BYNP                                      DSK01140
      RETURN                                                            DSK01150
      ENTRY WDISSF(NTRACK)                                              DSK01160
      WRITE(26'NTRACK) STNP                                             DSK01170
      RETURN                                                            DSK01180
      ENTRY RDISSF(NTRACK)                                              DSK01190
      READ(26'NTRACK,ERR=500) STNP                                      DSK01200
      RETURN                                                            DSK01210
      ENTRY WDISLF(NTRACK)                                              DSK01220
      WRITE(29'NTRACK) FLOAD                                            DSK01230
      RETURN                                                            DSK01240
      ENTRY RDISLF(NTRACK)                                              DSK01250
```

```
      READ(29'NTRACK,ERR=500) FLOAD
      RETURN
  500 WRITE(6,1000)
 1000 FORMAT('0',10X,'***** A MACHINE ERROR HAS OCCURRED DURING
     1DISK OPERATION, EXECUTIONS STOPS *****')
      STOP
      END
C
C***********************************************************************
C***********************************************************************
C
      SUBROUTINE NDPROP (XI,ETA,ZETA,IFLAG)
C
C     THIS SUBROUTINE FORMS THE ELASTIC MATRIX FOR NODAL PROPERTIES
C
      IMPLICIT REAL*8(A-H,O-Z)
      COMMON /NB1/ NEL,NDPT,NPEL,NDF,NEQ,NN,MM,NS,NCOUNT,NST,NSTF,NCLD,N
     1PBC,NPBC2,NBCL,NBCH,NMAT,NCON21,NDFP,NLINE,NREC7,KEL,NTLD,NBLD,NSL
     2D
      COMMON /B2/ ELAST(6,6),CCORD(3),COREL(20,3),UJNT(3),SSSS1(7),SSNN1
     1(7),ALPHA
      COMMON /B3/ AK1(60,60),AK2(60,60),AK3(60,60),RB1(60),RB2(60),RB3(6
     10)
      DIMENSION ELDATE(20,3), COL(20)
      EQUIVALENCE (RB2(1),ELDATE(1)), (AK3(1,33),COL(1))
      CALL SHAPE (XI,ETA,ZETA,COL)
      IF (IFLAG.EQ.1) GO TO 130
      DO 110 L=1,6
      DO 110 J=1,6
  110 ELAST(L,J) = 0.0D0
      E = 0.0D0
      PR = 0.0D0
      DO 120 I=1,NPEL
      E = E+COL(I)*ELDATE(I,1)
  120 PR = PR+COL(I)*ELDATE(I,2)
      ER = E/((1.0D0+PR)*(1.0D0-2.0D0*PR))
      EK = ER*(1.0D0-PR)
      EL = ER*PR
      ELAST(1,1) = EK
      ELAST(1,2) = EL
      ELAST(1,3) = EL
      ELAST(2,1) = EL
      ELAST(2,2) = EK
      ELAST(2,3) = EL
      ELAST(3,1) = EL
      ELAST(3,2) = EL
      ELAST(3,3) = EK
      ELAST(4,4) = ER*(1.0D0-2.0D0*PR)/2.0D0
```

```
      ELAST(5,5) = ELAST(4,4)                                           NDP00350
      ELAST(6,6) = ELAST(4,4)                                           NDP00360
      IF (IFLAG.EQ.0) RETURN                                            NDP00370
  130 ALPHA = 0.0D0                                                     NDP00380
      DO 140 I=1,NPEL                                                   NDP00390
  140 ALPHA = ALPHA+COL(I)*ELDATE(I,3)                                  NDP00400
      RETURN                                                            NDP00410
      END                                                               NDP00420
C
C***********************************************************************
C***********************************************************************
C
      SUBROUTINE SYMINV (A,N,B,C,IFLG)                                  SMV00010
C
C     THIS SUBROUTINE INVERTS A SYMMETRIC MATRIX
C
      IMPLICIT REAL*8(A-H,O-Z)                                          SMV00020
      DIMENSION A(1), B(1), C(1)                                        SMV00030
      IFLG = 0                                                          SMV00040
      ZRO = 0.0D0                                                       SMV00050
      ABIGN = 1.0D15                                                    SMV00060
      TRACE = ZRO                                                       SMV00070
      DO 110 I=1,N                                                      SMV00080
      IAD = (I-1)*N+I                                                   SMV00090
      IF (A(IAD).GE.ABIGN) GO TO 110                                    SMV00100
      TRACE = TRACE+DABS(A(IAD))                                        SMV00110
  110 CONTINUE                                                          SMV00120
      APZRO = TRACE*1.0D-12                                             SMV00130
      IF (APZRO.GT.1.0D-20) APZRO=1.0D-20                               SMV00140
      LP = N                                                            SMV00150
      DO 120 I=2,LP                                                     SMV00160
      DO 120 J=I,LP                                                     SMV00170
      ICOL = J+LP*(I-2)                                                 SMV00180
      IROW = I+(J-1)*LP-1                                               SMV00190
      IF (A(ICOL).EQ.A(IROW)) GO TO 120                                 SMV00200
      A(ICOL) = 0.5D0*(A(ICOL)+A(IROW))                                 SMV00210
      A(IROW) = A(ICOL)                                                 SMV00220
  120 CONTINUE                                                          SMV00230
      DO 140 I=1,N                                                      SMV00240
      B(I) = ZRO                                                        SMV00250
      II = (I-1)*N                                                      SMV00260
      DO 130 J=1,N                                                      SMV00270
      JJ = II+J                                                         SMV00280
      IF (A(JJ).GE.ABIGN) GO TO 130                                     SMV00290
      B(I) = B(I)+DABS(A(JJ))                                           SMV00300
  130 CONTINUE                                                          SMV00310
  140 CONTINUE                                                          SMV00320
      ANR = ZRO                                                         SMV00330
```

```
      DO 150 I=1,N                                                      SMV00340
  150 ANR = DMAX1(ANR,B(I))                                             SMV00350
      DO 220 I=1,N                                                      SMV00360
      NR = (I-1)*N                                                      SMV00370
      DO 160 J=1,N                                                      SMV00380
      K = NR+J                                                          SMV00390
  160 B(J) = A(K)                                                       SMV00400
      D = B(I)                                                          SMV00410
      IF (D.EQ.ZRO) GO TO 270                                           SMV00420
      IF (DABS(D).LT.APZRO) IFLG=2                                      SMV00430
      DO 170 J=1,N                                                      SMV00440
  170 C(J) = -B(J)/D                                                    SMV00450
      L = 1                                                             SMV00460
      DO 200 J=1,N                                                      SMV00470
      M = L                                                             SMV00480
      DO 190 K=J,N                                                      SMV00490
      DB = DABS(B(J))                                                   SMV00500
      IF (DB.LE.1.0D-40) GO TO 180                                      SMV00510
      DC = DABS(C(K))                                                   SMV00520
      IF (DC.LE.1.0D-40) GO TO 180                                      SMV00530
      IF ((DB.LE.APZRO).AND.(DC.LE.APZRO)) GO TO 180                    SMV00540
      A(L) = A(L)+B(J)*C(K)                                             SMV00550
  180 CONTINUE                                                          SMV00560
      A(M) = A(L)                                                       SMV00570
      M = M+N                                                           SMV00580
  190 L = L+1                                                           SMV00590
  200 L = L+J                                                           SMV00600
      C(I) = -1.0D0/D                                                   SMV00610
      M = I                                                             SMV00620
      DO 210 J=1,N                                                      SMV00630
      K = NR+J                                                          SMV00640
      A(K) = C(J)                                                       SMV00650
      A(M) = C(J)                                                       SMV00660
  210 M = M+N                                                           SMV00670
  220 CONTINUE                                                          SMV00680
      NS = N*N                                                          SMV00690
      DO 230 J=1,NS                                                     SMV00700
  230 A(J) = -A(J)                                                      SMV00710
      DO 250 I=1,N                                                      SMV00720
      B(I) = ZRO                                                        SMV00730
      II = (I-1)*N                                                      SMV00740
      DO 240 J=1,N                                                      SMV00750
      JJ = II+J                                                         SMV00760
      IF (A(JJ).EQ.1.0D0) GO TO 240                                     SMV00770
      B(I) = B(I)+DABS(A(JJ))                                           SMV00780
  240 CONTINUE                                                          SMV00790
  250 CONTINUE                                                          SMV00800
      AINR = ZRO                                                        SMV00810
```

```
      DO 260 I=1,N                                                      SMV00820
  260 AINR = DMAX1(AINR,B(I))                                           SMV00830
      IF (AINR.LT.1.0D-15) AINR=1.0D0/ANR                               SMV00840
      CNBR = ANR*AINR                                                   SMV00850
      WRITE (6,280) CNBR,ANR,AINR                                       SMV00860
      RETURN                                                            SMV00870
  270 IFLG = 1                                                          SMV00880
      RETURN                                                            SMV00890
  280 FORMAT (5X,'CONDITION NUMBER ',5X,1PD25.16,5X,2(1PD25.16))        SMV00900
      END                                                               SMV00910
C
C***********************************************************************
C***********************************************************************
C
      SUBROUTINE DLOAD                                                  DLD00010
C
C     THIS SUBROUTINE FORMS TOTAL LOAD VECTORS
C
      IMPLICIT REAL*8(A-H,O-Z)                                          DLD00020
      COMMON /NB1/ NEL,NDPT,NPEL,NDF,NEQ,NN,MM,NS,NCOUNT,NST,NSTF,NCLD,NDLD00030
     1PBC,NPBC2,NBCL,NBCH,NMAT,NCON21,NDFP,NLINE,NREC7,KEL,NTLD,NBLD,NSLDLD00040
     2D                                                                 DLD00050
      COMMON /B1/ ELDAT(10,3),CLOAD(4),BCON(3)                          DLD00060
      COMMON /B12/ ELDATN(3),BYNP(3),STNP(3),THNP,FLOAD(3)              DLD00070
      COMMON /B3/ AK1(60,60),AK2(60,60),AK3(60,60),RB1(60),RB2(60),RB3(6DLD00080
     10)                                                                DLD00090
      COMMON /CNTRL/ NLOAD                                              DLD00100
      DIMENSION A(60,60), S(60,60), R1(60), R2(60)                      DLD00110
      EQUIVALENCE (A(1,1),AK1(1,1)), (S(1,1),AK2(1,1)), (R1(1),RB1(1)), DLD00120
     1(R2(1),RB2(1))                                                    DLD00130
      NTKR(I,J) = (I-1)*NN+J                                            DLD00140
      NRTK(I,J) = (I-1)*NS3+J                                           DLD00150
      NTRK(I,J) = (I-1)*NDPT+J                                          DLD00160
      NS3 = NS/3                                                        DLD00170
      ZRO = 0.0D0                                                       DLD00180
      DO 130 IL=1,NLOAD                                                 DLD00190
      IF (NCLD.NE.0) REWIND 14                                          DLD00200
      IF (NCLD.EQ.0) GO TO 10                                           DLD00210
      READ (14) CLOAD                                                   DLD00220
      INDEX = 1                                                         DLD00230
      INCH = CLOAD(NDFP)*NDF                                            DLD00240
   10 DO 90 I=1,NN                                                      DLD00250
      DO 20 J=1,NS                                                      DLD00260
   20 R1(J) = ZRO                                                       DLD00270
      IF (IABS(NTLD)+IABS(NBLD)+IABS(NSLD).EQ.0) GO TO 50               DLD00280
      DO 40 J=1,NS3                                                     DLD00290
      NTK1 = NRTK(I,J)                                                  DLD00300
      IF (NTK1.GT.NDPT) GO TO 50                                        DLD00310
```

```
      NTK = NTRK(IL,NTK1)                                               DLD00320
      CALL RDISLF (NTK)                                                 DLD00330
      DO 30 K=1,NDF                                                     DLD00340
      IR = (J-1)*NDF+K                                                  DLD00350
   30 R1(IR) = FLOAD(K)                                                 DLD00360
   40 CONTINUE                                                          DLD00370
   50 IF (NCLD.EQ.0) GO TO 80                                           DLD00380
      IH = I*NS                                                         DLD00390
   60 IF (INCH.GT.IH) GO TO 80                                          DLD00400
      IR1 = INCH-(I-1)*NS-NDF                                           DLD00410
      DO 70 K=1,NDF                                                     DLD00420
      IR = IR1+K                                                        DLD00430
   70 R1(IR) = CLOAD(K)                                                 DLD00440
      IF (INDEX.EQ.NCLD.AND.NLOAD.EQ.0) GO TO 100                       DLD00450
      IF (INDEX.EQ.NCLD) GO TO 80                                       DLD00460
      READ (14) CLOAD                                                   DLD00470
      INDEX = INDEX+1                                                   DLD00480
      INCH = CLOAD(NDFP)*NDF                                            DLD00490
      GO TO 60                                                          DLD00500
   80 NTK = NTKR(IL,I)                                                  DLD00510
      CALL WDSKR1 (NTK)                                                 DLD00520
   90 CONTINUE                                                          DLD00530
      GO TO 130                                                         DLD00540
  100 NTK = NTKR(IL,I)                                                  DLD00550
      CALL WDSKR1 (NTK)                                                 DLD00560
      I1 = I+1                                                          DLD00570
      IF (I1.GT.NN) RETURN                                              DLD00580
      DO 120 K=I1,NN                                                    DLD00590
      DO 110 L=1,NS                                                     DLD00600
  110 R1(L) = ZRO                                                       DLD00610
      NTK = NTRK(IL,K)                                                  DLD00620
      CALL WDSKR1 (NTK)                                                 DLD00630
  120 CONTINUE                                                          DLD00640
  130 CONTINUE                                                          DLD00650
      RETURN                                                            DLD00660
      END                                                               DLD00670
C
C***********************************************************************
C***********************************************************************
C
      SUBROUTINE SORT                                                   SRT00010
C
C     FOR ALL LOAD CONDITIONS THIS SUBROUTINE SORTS NODAL VALUES INTO
C     ELEMENT STORAGE
C
      IMPLICIT REAL*8(A-H,O-Z)                                          SRT00020
      COMMON /NB1/ NEL,NDPT,NPEL,NDF,NEQ,NN,MM,NS,NCOUNT,NST,NSTF,NCLD,NSRT00030
     1PBC,NPBC2,NBCL,NBCH,NMAT,NCON21,NDFP,NLINE,NREC7,KEL,NTLD,NBLD,NSLSRT00040
```

```
     2D                                                                 SRT00050
      COMMON /NB2/ NCON(20),NBC(4)                                      SRT00060
      COMMON /B2/ ELAST(6,6),COORD(3),COREL(20,3),UJNT(3),SSSS1(7),SSNN1SRT00070
     1(7),ALPHA                                                         SRT00080
      COMMON /B12/ ELDATN(3),BYNP(3),STNP(3),THNP,FLOAD(3)              SRT00090
      COMMON /CNTRL/ NLOAD                                              SRT00100
      DIMENSION ELDATE(20,3), BYEP(20,3), STEP(20,3), TEEL(20)          SRT00110
      EQUIVALENCE (COREL(1,1),ELDATE(1,1),BYEP(1,1),STEP(1,1),TEEL(1))  SRT00120
      NTRK(I,J) = (I-1)*NDPT+J                                          SRT00130
      REWIND 9                                                          SRT00140
      REWIND 19                                                         SRT00150
      REWIND 23                                                         SRT00160
      REWIND 24                                                         SRT00170
      REWIND 27                                                         SRT00180
      REWIND 28                                                         SRT00190
C
C     SORT COORDINATES
C
      DO 130 I=1,NEL                                                    SRT00200
      READ (9) NCON                                                     SRT00210
      DO 110 J=1,NPEL                                                   SRT00220
      NTRACK = NCON(J)                                                  SRT00230
      CALL RDISKC (NTRACK)                                              SRT00240
      DO 110 K=1,NDF                                                    SRT00250
      COREL(J,K) = COORD(K)                                             SRT00260
  110 CONTINUE                                                          SRT00270
      WRITE (19) COREL                                                  SRT00280
C
C     SORT NODAL PROPERTIES
C
      IF (NMAT.GT.0) GO TO 130                                          SRT00290
      DO 120 J=1,NPEL                                                   SRT00300
      NTRACK = NCON(J)                                                  SRT00310
      CALL RDISKP (NTRACK)                                              SRT00320
      DO 120 K=1,NDF                                                    SRT00330
      ELDATE(J,K) = ELDATN(K)                                           SRT00340
  120 CONTINUE                                                          SRT00350
      WRITE (23) ELDATE                                                 SRT00360
  130 CONTINUE                                                          SRT00370
      IF (IABS(NTLD)+IABS(NBLD)+IABS(NSLD).EQ.0) RETURN                 SRT00380
      DO 200 IL=1,NLOAD                                                 SRT00390
      REWIND 9                                                          SRT00400
      DO 190 I=1,NEL                                                    SRT00410
      READ (9) NCON                                                     SRT00420
      IF (NBLD.EQ.0) GO TO 150                                          SRT00430
C
C     SORT BODY FORCES
C
```

```
      DO 140 J=1,NPEL                                                   SRT00440
      NTRACK = NCON(J)                                                  SRT00450
      NTK = NTRK(IL,NTRACK)                                             SRT00460
      CALL RDISBF (NTK)                                                 SRT00470
      DO 140 K=1,NDF                                                    SRT00480
      BYEP(J,K) = BYNP(K)                                               SRT00490
  140 CONTINUE                                                          SRT00500
      WRITE (27) BYEP                                                   SRT00510
  150 IF (NSLD.EQ.0) GO TO 170                                          SRT00520
C
C     SORT INITIAL DISPLACEMENTS
C
      DO 160 J=1,NPEL                                                   SRT00530
      NTRACK = NCON(J)                                                  SRT00540
      NTK = NTRK(IL,NTRACK)                                             SRT00550
      CALL RDISSF (NTK)                                                 SRT00560
      DO 160 K=1,NDF                                                    SRT00570
      STEP(J,K) = STNP(K)                                               SRT00580
  160 CONTINUE                                                          SRT00590
      WRITE (28) STEP                                                   SRT00600
  170 IF (NTLD.EQ.0) GO TO 190                                          SRT00610
C
C     SORT NODAL TEMPERATURES
C
      DO 180 J=1,NPEL                                                   SRT00620
      NTRACK = NCON(J)                                                  SRT00630
      NTK = NTRK(IT,NTRACK)                                             SRT00640
      CALL RDISKT (NTK)                                                 SRT00650
      TEEL(J) = THNP                                                    SRT00660
  180 CONTINUE                                                          SRT00670
      WRITE (24) TEEL                                                   SRT00680
  190 CONTINUE                                                          SRT00690
  200 CONTINUE                                                          SRT00700
      RETURN                                                            SRT00710
      END                                                               SRT00720
```

APPENDIX H

COMPUTER LISTING OF POSTPROCESSING PROGRAM

```
                    POST PROCESSING PROGRAM                             MAIN0010
C     THIS PROGRAM PROCESSES INFORMATION FROM THE STRESS ANALYSIS.      MAIN0020
C     PRESENTATION OF RESULTS IS PROVIDED IN A CONCISE FORMAT AND       MAIN0030
C     PLCTS OF INFORMATION ARE PROVIDED. LT. R. B. BUBECK, NAVAL        MAIN0040
C     PCSTGRADUATE SCHOOL, 1975                                         MAIN0041
C                                                                       MAIN0050
C                                                                       MAIN0060
C     INPUT DATA                                                        MAIN0070
C     TITLE CARD(3A4)                                                   MAIN0080
C     THE TITLE APPEARS ON THE RESULT PRESENTATION AND THE PLOTS.       MAIN0090
C     PROBLEM PARAMETERS: SIX INTEGERS,ONE FLOATING POINT (6I4,F10.0)   MAIN0100
C     NLCAD1..NUMBER OF FIRST LOAD TO BE PROCESSED                      MAIN0110
C     NLOAD2..NUMBER OF LAST LOAD TO BE PROCESSED                       MAIN0111
C     NPT.....NUMBER OF POINTS TO BE PROCESSED                          MAIN0120
C     NCORD...COORDINATE TO BE USED FOR PLOTS(1 - X,2 - Y,3 - Z)        MAIN0130
C     NDPT....TOTAL NUMBER OF POINTS IN THE MESH                        MAIN0140
C     NPLOT...PLOT INDICATOR 0 - PRINTER PLOT, 1 - CALCOMP PLOT         MAIN0150
C             PROVIDED.                                                 MAIN0160
C     TREF....ANY REFERENCE TEMPERATURE FOR ZERO STRESS CONDITION       MAIN0161
C                                                                       MAIN0170
C                                                                       MAIN0180
C     NODAL POINT INFORMATION(20I4)                                     MAIN0190
C     NCCN - THE NODAL NUMBERS OF POINTS TO BE PROCESSED.               MAIN0200
C                                                                       MAIN0210
C                                                                       MAIN0220
      REAL*8 SSSS1(7),COARD(3),THNP                                     MAIN0221
      DIMENSION NCON(20),RNG(4)                                         MAIN0230
      DIMENSION SSS(7),X(20),Y(20),Z(20),SX(20),SY(20),SZ(20)           MAIN0240
      EQUIVALENCE (SX(1),SSSS(1,1)),(SY(1),SSSS(1,2)),(SZ(1),SSSS(1,3)),MAIN0250
     1(X(1),COORD(1,1)),(Y(1),COORD(1,2)),(Z(1),COORD(1,3))             MAIN0260
      COMMON /MTRX/ SSSS(20,6),COORD(20,3),TEMP(20),TITLE(3)            MAIN0270
      DATA STOP/'STOP'/                                                 MAIN0280
      DEFINE FILE 8(2800,  14,U,I8),                                    MAIN0290
     1            9( 278,   6,U,I9),                                    MAIN0300
     2           10(2800,   2,U,I10)                                    MAIN0310
    1 READ(5,1000) TITLE                                                MAIN0320
      IF(TITLE(1).EQ.STOP) STOP                                         MAIN0330
      WRITE(6,1800)                                                     MAIN0340
      WRITE(6,1500) TITLE                                               MAIN0350
      READ(5,1110) NLOAD1,NLOAD2,NPT,NCORD,NDPT,NPLOT,TREF              MAIN0360
      READ(5,1100) (NCON(I),I=1,NPT)                                    MAIN0370
      WRITE(6,1101) NLOAD1,NLOAD2,NPT,NCORD,NDPT,NPLOT,(NCON(I),I=1,NPT)MAIN0380
```

```
      DO 900 IL=NLOAD1,NLOAD2                                          MAIN0390
      SMAX=-1.0E20                                                     MAIN0400
      SMIN=+1.0E20                                                     MAIN0410
      TMAX=-1.0E20                                                     MAIN0420
      TMIN=+1.0E20                                                     MAIN0430
      XMAX=-1.0E20                                                     MAIN0440
      XMIN=+1.0E20                                                     MAIN0450
      REWIND 11                                                        MAIN0460
      DO 10 I=1,IL                                                     MAIN0470
   10 READ(11) TIME                                                    MAIN0480
      DO 50 I=1,NPT                                                    MAIN0490
      NODE=NCON(I)                                                     MAIN0500
      READ(9'NODE) COARD                                               MAIN0510
      DO 50 J=1,3                                                      MAIN0520
   50 COORD(I,J)=DABS(COARD(J))                                        MAIN0530
      DO 100 I=1,NPT                                                   MAIN0540
      NODE=NCON(I)                                                     MAIN0550
      NTK=(IL-1)*NDPT+NODE                                             MAIN0560
      READ(8'NTK) SSSS1                                                MAIN0570
      READ(10'NTK) THNP                                                MAIN0580
      DO 110 J=1,6                                                     MAIN0590
  110 SSSS(I,J)=SSSS1(J)/SSSS1(7)                                      MAIN0600
      TEMP(I)=THNP+TREF                                                MAIN0610
      IF(NPLOT.NE.0) GO TO 100                                         MAIN0620
      SMAX=AMAX1(SMAX,SSSS(I,1),SSSS(I,2),SSSS(I,3))                   MAIN0630
      SMIN=AMIN1(SMIN,SSSS(I,1),SSSS(I,2),SSSS(I,3))                   MAIN0640
      TMAX=AMAX1(TMAX,TEMP(I))                                         MAIN0650
      TMIN=AMIN1(TMIN,TEMP(I))                                         MAIN0660
      XMAX=AMAX1(XMAX,COORD(I,NCORD))                                  MAIN0670
      XMIN=AMIN1(XMIN,COORD(I,NCORD))                                  MAIN0680
  100 CONTINUE                                                         MAIN0690
      WRITE(6,1400) TIME                                               MAIN0700
      DO 200 I=1,NPT                                                   MAIN0710
      DO 205 J=1,7                                                     MAIN0720
  205 SSS(J)=SSSS(I,J)                                                 MAIN0730
      CALL PRNSTR(SSS,S1,S2,S3)                                        MAIN0740
      WRITE(6,1205) COORD(I,1),SSSS(I,1),SSSS(I,4),S1                  MAIN0750
      WRITE(6,1300) NCON(I),COORD(I,2),TEMP(I),SSSS(I,2),SSSS(I,5),S2  MAIN0760
  200 WRITE(6,1200) COORD(I,3),SSSS(I,3),SSSS(I,6),S3                  MAIN0770
      IF(NCORD.LE.3) GO TO 300                                         MAIN0780
      CALL RADIAL(NPT,NCORD,NPLOT,NCON,TIME)                           MAIN0790
      GO TO 860                                                        MAIN0800
  300 IF(NPLOT.NE.0) GO TO 850                                         MAIN0810
      RNG(1)=XMAX                                                      MAIN0820
      RNG(2)=XMIN                                                      MAIN0830
      RNG(3)=SMAX                                                      MAIN0840
      RNG(4)=SMIN                                                      MAIN0850
      GO TO (600,700,800),NCORD                                        MAIN0860
```

```
  600 WRITE(6,1801)                                                MAIN0870
      CALL UTPLOT(X,SX,NPT,RNG,1,1)                                MAIN0880
      CALL UTPLOT(X,SY,NPT,RNG,1,2)                                MAIN0890
      CALL UTPLOT(X,SZ,NPT,RNG,1,3)                                MAIN0900
      WRITE(6,1802)                                                MAIN0910
      RNG(3)=TMAX                                                  MAIN0920
      RNG(4)=TMIN                                                  MAIN0930
      CALL UTPLOT(X,TEMP,NPT,RNG,1,0)                              MAIN0940
      GO TO 850                                                    MAIN0950
  700 WRITE(6,1803)                                                MAIN0960
      CALL UTPLOT(Y,SX,NPT,RNG,1,1)                                MAIN0970
      CALL UTPLOT(Y,SY,NPT,RNG,1,2)                                MAIN0980
      CALL UTPLOT(Y,SZ,NPT,RNG,1,3)                                MAIN0990
      WRITE(6,1804)                                                MAIN1000
      RNG(3)=TMAX                                                  MAIN1010
      RNG(4)=TMIN                                                  MAIN1020
      CALL UTPLOT(Y,TEMP,NPT,RNG,1,0)                              MAIN1030
      GO TO 850                                                    MAIN1040
  800 WRITE(6,1805)                                                MAIN1050
      CALL UTPLOT(Z,SX,NPT,RNG,1,1)                                MAIN1060
      CALL UTPLOT(Z,SY,NPT,RNG,1,2)                                MAIN1070
      CALL UTPLOT(Z,SZ,NPT,RNG,1,3)                                MAIN1080
      WRITE(6,1806)                                                MAIN1090
      RNG(3)=TMAX                                                  MAIN1100
      RNG(4)=TMIN                                                  MAIN1110
      CALL UTPLOT(Z,TEMP,NPT,RNG,1,0)                              MAIN1120
  850 IF(NPLOT.NE.0) CALL GRAPH(NPT,NCORD,TIME)                    MAIN1130
  860 P1MAX=-1.0E20                                                MAIN1140
      P1MIN=+1.0D20                                                MAIN1150
      DO 875 I=1,NDPT                                              MAIN1160
      NTK=(IL-1)*NDPT+I                                            MAIN1170
      READ(8'NTK) SSSS1                                            MAIN1180
      DO 880 J=1,7                                                 MAIN1190
  880 SSS(J)=SSSS1(J)/SSSS1(7)                                     MAIN1200
      CALL PRNSTR(SSS,S1,S2,S3)                                    MAIN1210
      P2MAX=AMAX1(P1MAX,S1,S2,S3)                                  MAIN1220
      P2MIN=AMIN1(P1MIN,S1,S2,S3)                                  MAIN1230
      IF(P1MAX.NE.P2MAX) IMAX=I                                    MAIN1240
      IF(P1MIN.NE.P2MIN) IMIN=I                                    MAIN1250
      P1MAX=P2MAX                                                  MAIN1260
      P1MIN=P2MIN                                                  MAIN1270
  875 CONTINUE                                                     MAIN1280
      WRITE(6,1900) TIME,P1MAX,IMAX,P1MIN,IMIN                     MAIN1290
  900 CONTINUE                                                     MAIN1300
      GO TO 1                                                      MAIN1310
 1000 FORMAT(3A4)                                                  MAIN1320
 1100 FORMAT(20I4)                                                 MAIN1330
 1110 FORMAT(6I4,F10.0)                                            MAIN1331
```

```
1101 FORMAT('0',14X,'NUMBER OF LOADS TO BE PROCESSED = ',2I5,//,15X,'NU
    1MBER OF POINTS TO BE PROCESSED = ',I5,//,15X,'COORDINATE TO BE USE
    2D = ',I5,//,15X,'TOTAL NUMBER OF NODES IN MESH = ',I5,//,15X,'VALU
    3E OF NPLOT = ',I5,//,15X,'NODAL NUMBERS TO BE PROCESSED',//,20X,
    420I5,///)
1200 FORMAT(' ',15X,1P1G18.6,18X,1P3G18.6)
1205 FORMAT('0',15X,1P1G18.6,18X,1P3G18.6)
1300 FORMAT(' ',10X,I5,1P5G18.6)
1400 FORMAT('1',38X,'TEMPERATURE AND STRESS VALUES, TIME = ',F10.5,//,1
    13X,'NODE',5X,'COORDINATE',8X,'TEMPERATURE',10X,'SIGMA',14X,'TAU',1
    21X,'PRINCIPAL',/,25X,'X/Y/Z',31X,'X/Y/Z',12X,'XY/YZ/ZX',09X,'STRES
    3S',/)
1500 FORMAT('1',25X,3A4,//)
1700 FORMAT('1',5X,1P3G20.6)
1701 FORMAT(' ',5X,1P5G20.6)
1800 FORMAT('1')
1801 FORMAT('1',53X,'STRESS VERSUS X COORDINATE',//,44X,'SIGMA X - "."
    1,SIGMA Y - "+" , SIGMA Z - "*"',///)
1802 FORMAT('1',51X,'TEMPERATURE VERSUS X COORDINATE',////)
1803 FORMAT('1',53X,'STRESS VERSUS Y COORDINATE',//,44X,'SIGMA X - "."
    1,SIGMA Y - "+" , SIGMA Z - "*"',///)
1804 FORMAT('1',51X,'TEMPERATURE VERSUS Y COORDINATE',////)
1805 FORMAT('1',53X,'STRESS VERSUS Z COORDINATE',//,44X,'SIGMA X - "."
    1,SIGMA Y - "+" , SIGMA Z - "*"',///)
1806 FORMAT('1',51X,'TEMPERATURE VERSUS Z COORDINATE',////)
1900 FORMAT('0',/////,25X,'AT TIME = ',F10.5,'PRINCIPAL STRESS ARE : ',
    1//,25X,'MAXIMUM PRINCIPAL STRESS = ',1P1G20.6,' AT NODE',I5,//,25X
    2,'MINIMUM PRINCIPAL STRESS = ',1P1G20.6,' AT NODE',I5)
     END
C
C***********************************************************************
C***********************************************************************
C
      SUBROUTINE PRNSTR(STR,S1,S2,S3)
      COMPLEX*8 AC,BC,CC
      DIMENSION STR(1)
      DATA CC/(0.0D0,1.732050808D0)/,CR/0.333333333333D0/
      SX=STR(1)
      SY=STR(2)
      SZ=STR(3)
      TXY=STR(4)
      TYZ=STR(5)
      TXZ=STR(6)
      PHI=-(SX+SY+SZ)
      PHII=PHI
      THETA=SX*SY+SY*SZ+SX*SZ-(TXY*TXY+TXZ*TXZ+TXY*TXY)
      PSI=-(SX*SY*SZ+2.0*TYZ*TXZ*TXY-(SX*TYZ*TYZ+SY*TXZ*TXZ+SZ*TXY*TXY))
      A=THETA-PHI*PHI/3.0D0
```

```
      B=2.0D0*PHI*PHI*PHI/27.0D0-PHI*THETA/3.0D0+PSI
      AB=B*B/4.0D0+A*A*A/27.0D0
      IF(AB.LT.0.0D0) GO TO 100
      AB=SQRT(AB)
      BPAB=-B/2.0D0+AB
      BMAB=-B/2.0D0-AB
      IF(BPAB.LT.0.0D0) GO TO 10
      AC=BPAB**CR
      GO TO 20
   10 AC=-((-BPAB)**CR)
   20 IF(BMAB.LT.0.0D0) GO TO 30
      BC=BMAB**CR
      GO TO 40
   30 BC=-((-BMAB)**CR)
   40 S1=REAL(AC+BC)
      S2=REAL(-(AC+BC)/2.0D0+CC*(A-B)/2.0D0)
      S3=REAL(-(AC+BC)/2.0D0-CC*(A-B)/2.0D0)
      S1=S1-PHII/3.0D0
      S2=S2-PHII/3.0D0
      S3=S3-PHII/3.0D0
      RETURN
  100 AB=(-B/2.0D0)/SQRT(-A*A*A/27.0)
      PHI=ARCOS(AB)
      FACT=2.0D0*SQRT(-A/3.0)
      PHI3=PHI/3.0D0
      S1=FACT*COS(PHI3)
      S2=FACT*COS(PHI3+2.094395103)
      S3=FACT*COS(PHI3+4.18879206)
      S1=S1-PHII/3.0D0
      S2=S2-PHII/3.0D0
      S3=S3-PHII/3.0D0
      RETURN
      END
C
C***********************************************************************
C***********************************************************************
C
      SUBROUTINE GRAPH(NPT,NCORD,TIME)
      REAL*4 BCD(4)/ZCA000000,Z5C000000,ZCC000000,ZCB000000/,
     1SP(4)/0.04,0.06,0.04,0.05/
      COMMON /MTRX/ SSSS(20,6),COORD(20,3),TEMP(20),TITLE(3)
      DIMENSION TITLEX(4),TITLYL(2),TITLYR(3),BCDI(1),
     1NCON(20),SX(20),SY(20),SZ(20),SXYZ(60),
     2X(20),Y(20),Z(20)
      EQUIVALENCE (SX(1),SSSS(1,1)),(SY(1),SSSS(1,2)),(SZ(1),SSSS(1,3)),
     1(X(1),COORD(1,1)),(Y(1),COORD(1,2)),(Z(1),CCORD(1,3))
      DATA TITLEX/'SPAT','IAL ','POSI','TION'/,TITLYL/'STRE','SS  '/,
     1TITLYR/'TEMP','ERAT','URE '/
```

```
C     INITIALIZE PLOTS                                                  GRPH0120
      CALL PLOTS                                                        GRPH0130
C     SCALE AXES                                                        GRPH0140
      IF(NCORD.EQ.1) CALL SCAL(X,NPT,5.0,XMIN,DX)                       GRPH0150
      IF(NCORD.EQ.2) CALL SCAL(Y,NPT,5.0,XMIN,DX)                       GRPH0160
      IF(NCORD.EQ.3) CALL SCAL(Z,NPT,5.0,XMIN,DX)                       GRPH0170
      NPT2=2*NPT                                                        GRPH0180
      NPT3=3*NPT                                                        GRPH0190
      DO 50 I=1,NPT                                                     GRPH0200
      SXYZ(I)=SX(I)                                                     GRPH0210
      SXYZ(I+NPT)=SY(I)                                                 GRPH0220
   50 SXYZ(I+NPT2)=SZ(I)                                                GRPH0230
      CALL SCAL(SXYZ,NPT3,6.0,SMIN,DS)                                  GRPH0240
      CALL SCAL(TEMP,NPT,3.0,TMIN,DT)                                   GRPH0250
C     DRAW AXES                                                         GRPH0260
      DO 75 I=1,NPT                                                     GRPH0270
      SX(I)=SXYZ(I)                                                     GRPH0280
      SY(I)=SXYZ(I+NPT)                                                 GRPH0290
   75 SZ(I)=SXYZ(I+NPT2)                                                GRPH0300
      XS=0.0                                                            GRPH0310
      YS=0.0                                                            GRPH0320
      IF(XMIN.LT.0.0) XS=ABS(XMIN)/DX                                   GRPH0330
      IF(SMIN.LT.0.0) YS=ABS(SMIN)/DS                                   GRPH0340
      CALL AXIS(XS,0.0,TITLYL,+6,6.0,90.0,SMIN,DS)                      GRPH0350
      CALL AXIS(XS,0.0,TITLYL,+6,6.0,90.0,SMIN,DS)                      GRPH0360
      CALL AXIS(0.0,YS,TITLEX,-16,5.0,0.0,XMIN,DX)                      GRPH0370
      CALL AXIS(0.0,YS,TITLEX,-16,5.0,0.0,XMIN,DX)                      GRPH0380
      XSP=XS+5.0                                                        GRPH0390
      CALL AXIS(XSP,YS,TITLYR,-11,3.0,90.0,TMIN,DT)                     GRPH0400
      CALL AXIS(XSP,YS,TITLYR,-11,3.0,90.0,TMIN,DT)                     GRPH0410
      DO 80 I=1,NPT                                                     GRPH0420
   80 TEMP(I)=TEMP(I)+YS                                                GRPH0430
C     PLOT VALUES                                                       GRPH0440
      GO TO (100,200,300),NCORD                                         GRPH0450
  100 CALL LINE(X,SX,NPT,1,1)                                           GRPH0460
      CALL LINE(X,SX,NPT,1,1)                                           GRPH0470
      CALL LINE(X,SY,NPT,1,1)                                           GRPH0480
      CALL LINE(X,SY,NPT,1,1)                                           GRPH0490
      CALL LINE(X,SZ,NPT,1,1)                                           GRPH0500
      CALL LINE(X,SZ,NPT,1,1)                                           GRPH0510
      CALL LINE(X,TEMP,NPT,1,1)                                         GRPH0520
      CALL LINE(X,TEMP,NPT,1,1)                                         GRPH0530
      CALL SYMBOL(.75,-.5,0.14,'SPATIAL POSITION - X COORDINATE',0.0,31)GRPH0540
      CALL SYMBOL(.75,-.5,0.14,'SPATIAL POSITION - X COORDINATE',0.0,31)GRPH0550
      GO TO 400                                                         GRPH0560
  200 CALL LINE(Y,SX,NPT,1,1)                                           GRPH0570
      CALL LINE(Y,SX,NPT,1,1)                                           GRPH0580
      CALL LINE(Y,SY,NPT,1,1)                                           GRPH0590
```

```
      CALL LINE(Y,SY,NPT,1,1)                                           GRPH0600
      CALL LINE(Y,SZ,NPT,1,1)                                           GRPH0610
      CALL LINE(Y,SZ,NPT,1,1)                                           GRPH0620
      CALL LINE(Y,TEMP,NPT,1,1)                                         GRPH0630
      CALL LINE(Y,TEMP,NPT,1,1)                                         GRPH0640
      CALL SYMBOL(.75,-.5,0.14,'SPATIAL POSITION - Y COORDINATE',0.0,31)GRPH0650
      CALL SYMBOL(.75,-.5,0.14,'SPATIAL POSITION - Y COORDINATE',0.0,31)GRPH0660
      GO TO 400                                                         GRPH0670
  300 CALL LINE(Z,SX,NPT,1,1)                                           GRPH0680
      CALL LINE(Z,SX,NPT,1,1)                                           GRPH0690
      CALL LINE(Z,SY,NPT,1,1)                                           GRPH0700
      CALL LINE(Z,SY,NPT,1,1)                                           GRPH0710
      CALL LINE(Z,SZ,NPT,1,1)                                           GRPH0720
      CALL LINE(Z,SZ,NPT,1,1)                                           GRPH0730
      CALL LINE(Z,TEMP,NPT,1,1)                                         GRPH0740
      CALL LINE(Z,TEMP,NPT,1,1)                                         GRPH0750
      CALL SYMBOL(.75,-.5,0.14,'SPATIAL POSITION - Z COORDINATE',0.0,31)GRPH0760
      CALL SYMBOL(.75,-.5,0.14,'SPATIAL POSITION - Z COORDINATE',0.0,31)GRPH0770
  400 DO 450 I=1,3                                                      GRPH0780
      BCDI(1)=BCD(I)                                                    GRPH0790
      DO 450 J=1,NPT                                                    GRPH0800
      XN=COORD(J,NCORD)-SP(I)                                           GRPH0810
      YN=SSSS(J,I)-SP(I)                                                GRPH0820
      CALL SYMBOL(XN,YN,0.14,BCDI,0.0,1)                                GRPH0830
  450 CALL SYMBOL(XN,YN,0.14,BCDI,0.0,1)                                GRPH0840
      BCDI(1)=BCD(4)                                                    GRPH0850
      DO 475 I=1,NPT                                                    GRPH0860
      XN=COORD(I,NCORD)-SP(4)                                           GRPH0870
      YN=TEMP(I)-SP(4)                                                  GRPH0880
      CALL SYMBOL(XN,YN,0.14,BCDI,0.0,1)                                GRPH0890
  475 CALL SYMBOL(XN,YN,0.14,BCDI,0.0,1)                                GRPH0900
      CALL SYMBOL(.5,-0.75,0.14,'SIGMA X - DIAMOND',0.0,17)             GRPH0910
      CALL SYMBOL(.5,-0.75,0.14,'SIGMA X - DIAMOND',0.0,17)             GRPH0920
      CALL SYMBOL(2.75,-.75,0.14,'SIGMA Y - ASTERISK',0.0,18)           GRPH0930
      CALL SYMBOL(2.75,-.75,0.14,'SIGMA Y - ASTERISK',0.0,18)           GRPH0940
      CALL SYMBOL(0.5,-1.0,0.14,'SIGMA Z - CROSS',0.0,15)               GRPH0950
      CALL SYMBOL(0.5,-1.0,0.14,'SIGMA Z - CROSS',0.0,15)               GRPH0960
      CALL SYMBOL(2.5,-1.00,0.14,'TEMPERATURE - SQUARE',0.0,20)         GRPH0970
      CALL SYMBOL(2.5,-1.00,0.14,'TEMPERATURE - SQUARE',0.0,20)         GRPH0980
      CALL SYMBOL(0.5,-1.25,0.07,'STRESS SCALE =       X 10   PSI/INCH',0GRPH0990
     1.0,35)                                                            GRPH1000
      CALL SYMBOL(0.5,-1.25,0.07,'STRESS SCALE =       X 10   PSI/INCH',0GRPH1010
     1.0,35)                                                            GRPH1020
      CALL ANN(DS,DSS,KEX)                                              GRPH1030
      PEX=KEX                                                           GRPH1040
      CALL NUMBER(1.375,-1.25,0.07,DSS,0.0,3)                           GRPH1050
      CALL NUMBER(1.375,-1.25,0.07,DSS,0.0,3)                           GRPH1060
      CALL NUMBER(1.880,-1.18,0.07,PEX,0.0,-1)                          GRPH1070
```

```
      CALL NUMBER(1.880,-1.18,0.07,PEX,0.0,-1)                          GRPH1080
      CALL SYMBOL(0.5,-1.5,0.07,'TEMPERATURE SCALE =       DEGREES/INCH GRPH1090
     1',0.0,38)                                                         GRPH1100
      CALL SYMBOL(0.5,-1.5,0.07,'TEMPERATURE SCALE =       DEGREES/INCH GRPH1110
     1',0.0,38)                                                         GRPH1120
      CALL NUMBER(1.6875,-1.5,0.07,DT,0.0,1)                            GRPH1130
      CALL NUMBER(1.6875,-1.5,0.07,DT,0.0,1)                            GRPH1140
      CALL SYMBOL(0.5,-1.75,0.07,'SPATIAL POSITION SCALE =      INCHES/ GRPH1150
     1INCH',0.0,42)                                                     GRPH1160
      CALL SYMBOL(0.5,-1.75,0.07,'SPATIAL POSITION SCALE =      INCHES/ GRPH1170
     1INCH',0.0,42)                                                     GRPH1180
      CALL NUMBER(1.9375,-1.75,0.07,DX,0.0,2)                           GRPH1190
      CALL NUMBER(1.9375,-1.75,0.07,DX,0.0,2)                           GRPH1200
      CALL SYMBOL(1.125,6.75,0.14,'STRESS AND TEMPERATURE',0.0,22)      GRPH1210
      CALL SYMBOL(1.125,6.75,0.14,'STRESS AND TEMPERATURE',0.0,22)      GRPH1220
      CALL SYMBOL(1.125,6.5,0.14,'VERSUS SPATIAL POSITION',0.0,23)      GRPH1230
      CALL SYMBOL(1.125,6.5,0.14,'VERSUS SPATIAL POSITION',0.0,23)      GRPH1240
      CALL SYMBOL(1.125,6.25,0.14,'TIME = ',0.0,7)                      GRPH1250
      CALL NUMBER(2.00,6.25,0.14,TIME,0.0,3)                            GRPH1260
      CALL SYMBOL(2.625,6.25,0.14,TITLE,0.0,12)                         GRPH1270
      CALL SYMBOL(1.125,6.25,0.14,'TIME = ',0.0,7)                      GRPH1280
      CALL NUMBER(2.00,6.25,0.14,TIME,0.0,3)                            GRPH1290
      CALL SYMBOL(2.625,6.25,0.14,TITLE,0.0,12)                         GRPH1300
      CALL PLOT(0.0,10.0,-3)                                            GRPH1310
      CALL PLOTE                                                        GRPH1320
      WRITE(6,1600)                                                     GRPH1330
 1600 FORMAT('0',25X,'*** GRAPHS HAVE BEEN DRAWN ***')                  GRPH1340
      RETURN                                                            GRPH1350
      END                                                               GRPH1360
C
C***********************************************************************
C***********************************************************************
C
      SUBROUTINE SCAL(Y,N,S,YMIN,DY)                                    SCAL0010
      DIMENSION Y(1)                                                    SCAL0020
C     DETERMINE SCALES                                                  SCAL0030
      YMAX=-1.0D20                                                      SCAL0040
      YMIN=+1.0D20                                                      SCAL0050
      DO 10 I=1,N                                                       SCAL0060
      VAL=Y(I)                                                          SCAL0070
      YMAX=AMAX1(VAL,YMAX)                                              SCAL0080
   10 YMIN=AMIN1(VAL,YMIN)                                              SCAL0090
      IF(ABS(YMAX).LT.1.0.AND.ABS(YMIN).LT.1.0) GO TO 15                SCAL0100
      TMAX=YMAX/10.0                                                    SCAL0110
      IMAX=INT(TMAX)+1                                                  SCAL0120
      TMAX=10.0*IMAX                                                    SCAL0130
      YMAX=TMAX                                                         SCAL0140
      TMIN=YMIN/10.0                                                    SCAL0150
```

```
      IMIN=INT(TMIN)                                                    SCAL0160
      TMIN=10.0*IMIN                                                    SCAL0170
      YMIN=TMIN                                                         SCAL0180
      IF(YMAX.GT.0.0.AND.YMIN.GT.0.0) GO TO 14                          SCAL0190
      IF(ABS(YMAX).GT.ABS(YMIN)) YMIN=SIGN(YMAX,YMIN)                   SCAL0200
      IF(ABS(YMIN).GT.ABS(YMAX)) YMAX=SIGN(YMIN,YMAX)                   SCAL0210
   14 DY=(YMAX-YMIN)/S                                                  SCAL0220
      GO TO 17                                                          SCAL0230
   15 TMAX=YMAX*100.0                                                   SCAL0240
      IMAX=INT(TMAX)+1                                                  SCAL0250
      TMAX=FLOAT(IMAX)/100.0                                            SCAL0260
      TMIN=YMIN*100.0                                                   SCAL0270
      IMIN=INT(TMIN)                                                    SCAL0280
      TMIN=FLOAT(IMIN)/100.0                                            SCAL0290
      DY=(TMAX-TMIN)/S                                                  SCAL0300
C     SCALE DATA                                                        SCAL0310
   17 SLOPE=1.0/DY                                                      SCAL0320
      DO 20 I=1,N                                                       SCAL0330
   20 Y(I)=SLOPE*(Y(I)-YMIN)                                            SCAL0340
      RETURN                                                            SCAL0350
      END                                                               SCAL0360
C
C***********************************************************************
C***********************************************************************
C
      SUBROUTINE RADIAL(NPT,NCORD,NPLOT,NCON,TIME)                      RADL0010
      REAL*4 BCD(4)/ZCA000000,Z5C000000,ZCC000000,ZCB000000/,           RADL0020
     1SP(4)/0.04,0.06,0.04,0.05/                                        RADL0030
      DIMENSION RNG(4),NCON(20),SXYZ(60),BCDI(1)                        RADL0040
      COMMON /MTRX/ SSSS(20,6),COORD(20,3),TEMP(20),TITLE(3)            RADL0050
      RAD=57.295779                                                     RADL0060
      SMAX=-1.0E20                                                      RADL0070
      SMIN=+1.0E20                                                      RADL0080
      TMAX=-1.0E20                                                      RADL0090
      TMIN=+1.0E20                                                      RADL0100
      XMAX=-1.0E20                                                      RADL0110
      XMIN=+1.0E20                                                      RADL0120
      WRITE(6,1000) TIME                                                RADL0130
C                                                                       RADL0140
C      SET UP COORDINATE INDICATORS                                     RADL0150
C                                                                       RADL0160
      NCORD1=NCORD/10                                                   RADL0170
      NCORD2=NCORD-NCORD1*10                                            RADL0180
      NCORD3=6-NCORD1-NCORD2                                            RADL0190
      IF(NCORD1*NCORD2.NE.2) GO TO 20                                   RADL0200
      NCORD4=4                                                          RADL0210
      NCORD5=5                                                          RADL0220
      NCORD6=6                                                          RADL0230
```

```
   20 IF(NCORD1*NCORD2.NE.3) GO TO 30                                   RADL0240
      NCORD4=6                                                          RADL0250
      NCORD5=4                                                          RADL0260
      NCORD6=5                                                          RADL0270
   30 IF(NCORD1*NCORD2.NE.6) GO TO 1                                    RADL0280
      NCORD4=5                                                          RADL0290
      NCORD5=6                                                          RADL0300
      NCORD6=4                                                          RADL0310
    1 DO 10 I=1,NPT                                                     RADL0320
C                                                                       RADL0330
C      CONVERT COORDINATES                                              RADL0340
C                                                                       RADL0350
      X=COORD(I,NCORD1)                                                 RADL0360
      Y=COORD(I,NCORD2)                                                 RADL0370
      R=SQRT(X*X+Y*Y)                                                   RADL0380
      THETA=ARCOS(X/R)                                                  RADL0390
      COORD(I,NCORD1)=R                                                 RADL0400
      COORD(I,NCORD2)=THETA*RAD                                         RADL0410
C                                                                       RADL0420
C      CONVERT STRESSES                                                 RADL0430
C                                                                       RADL0440
      SX=SSSS(I,NCORD1)                                                 RADL0450
      SY=SSSS(I,NCORD2)                                                 RADL0460
      SZ=SSSS(I,NCORD3)                                                 RADL0470
      TXY=SSSS(I,NCORD4)                                                RADL0480
      TYZ=SSSS(I,NCORD5)                                                RADL0490
      TXZ=SSSS(I,NCORD6)                                                RADL0500
      S=SIN(THETA)                                                      RADL0510
      C=COS(THETA)                                                      RADL0520
      S2=S*S                                                            RADL0530
      C2=C*C                                                            RADL0540
      CS2=2.0*C*S                                                       RADL0550
      SSSS(I,1)=SX*C2+SY*S2+TXY*CS2                                     RADL0560
      SSSS(I,2)=SX*S2+SY*C2-TXY*CS2                                     RADL0570
      SSSS(I,3)=SZ                                                      RADL0580
      SSSS(I,4)=(SY-SX)*C*S+TXY*(C2-S2)                                 RADL0590
      SSSS(I,5)=TXZ*C+TYZ*S                                             RADL0600
      SSSS(I,6)=TYZ*C-TXZ*S                                             RADL0610
      WRITE(6,1002) COORD(I,NCORD1),SSSS(I,1),SSSS(I,4)                 RADL0620
      WRITE(6,1001) NCON(I),COORD(I,NCORD2),TEMP(I),SSSS(I,2),SSSS(I,5) RADL0630
      WRITE(6,1003) COORD(I,NCORD3),SSSS(I,3),SSSS(I,6)                 RADL0640
      SMAX=AMAX1(SMAX,SSSS(I,1),SSSS(I,2),SSSS(I,3))                    RADL0650
      SMIN=AMIN1(SMIN,SSSS(I,1),SSSS(I,2),SSSS(I,3))                    RADL0660
      XMAX=AMAX1(R,XMAX)                                                RADL0670
      XMIN=AMIN1(XMIN,R)                                                RADL0680
      TMAX=AMAX1(TMAX,TEMP(I))                                          RADL0690
   10 TMIN=AMIN1(TMIN,TEMP(I))                                          RADL0700
      IF(NPLOT.NE.0) GO TO 100                                          RADL0710
```

```
      RNG(1)=XMAX                                                       RADL0720
      RNG(2)=XMIN                                                       RADL0730
      RNG(3)=SMAX                                                       RADL0740
      RNG(4)=SMIN                                                       RADL0750
      WRITE(6,2000)                                                     RADL0760
      DC 50 I=1,3                                                       RADL0770
      CALL UTPLOT(COORD(1,NCORD1),SSSS(1,I),NPT,RNG,1,I)                RADL0780
   50 CONTINUE                                                          RADL0790
      WRITE(6,2001)                                                     RADL0800
      RNG(3)=TMAX                                                       RADL0810
      RNG(4)=TMIN                                                       RADL0820
      CALL UTPLOT(COORD(1,NCORD1),TEMP,NPT,RNG,1,0)                     RADL0830
      RETURN                                                            RADL0831
  100 NPT2=NPT*2                                                        RADL0840
      NPT3=NPT*3                                                        RADL0850
      CALL PLCTS                                                        RADL0860
      CALL PLOT(1.0,0.0,-3)                                             RADL0870
C                                                                       RADL0880
C     SCALE AXES                                                        RADL0390
C                                                                       RADL0900
      CALL SCAL(COORD(1,NCORD1),NPT,5.0,XMIN,DX)                        RADL0910
      DO 150 I=1,NPT                                                    RADL0920
      SXYZ(I)=SSSS(I,1)                                                 RADL0930
      SXYZ(I+NPT)=SSSS(I,2)                                             RADL0940
  150 SXYZ(I+NPT2)=SSSS(I,3)                                            RADL0950
      CALL SCAL(SXYZ,NPT3,6.0,SMIN,DS)                                  RADL0960
      DO 175 I=1,NPT                                                    RADL0970
      SSSS(I,1)=SXYZ(I)                                                 RADL0980
      SSSS(I,2)=SXYZ(I+NPT)                                             RADL0990
  175 SSSS(I,3)=SXYZ(I+NPT2)                                            RADL1000
      CALL SCAL(TEMP,NPT,3.0,TMIN,DT)                                   RADL1010
      XS=0.0                                                            RADL1020
      YS=0.0                                                            RADL1030
      IF(XMIN.LT.0.0) XS=ABS(XMIN)/DX                                   RADL1040
      IF(SMIN.LT.0.0) YS=ABS(SMIN)/DS                                   RADL1050
      CALL AXIS(XS,0.0,'STRESS (PSI)',+12,6.0,90.0,SMIN,DS)             RADL1060
      CALL AXIS(XS,0.0,'STRESS (PSI)',+12,6.0,90.0,SMIN,DS)             RADL1070
      CALL AXIS(0.0,YS,'RADIUS (IN.)',-12,5.0,0.0,XMIN,DX)              RADL1080
      CALL AXIS(0.0,YS,'RADIUS (IN.)',-12,5.0,0.0,XMIN,DX)              RADL1090
      XSP=XS+5.0                                                        RADL1100
      CALL AXIS(XSP,YS,'TEMPERATURE (DEG.F)',-19,3.0,90.0,TMIN,DT)      RADL1110
      CALL AXIS(XSP,YS,'TEMPERATURE (DEG.F)',-19,3.0,90.0,TMIN,DT)      RADL1120
      DO 160 I=1,NPT                                                    RADL1130
  160 TEMP(I)=TEMP(I)+YS                                                RADL1140
      CALL LINE(COORD(1,NCORD1),SSSS(1,1),NPT,1,1)                      RADL1150
      CALL LINE(COORD(1,NCORD1),SSSS(1,1),NPT,1,1)                      RADL1160
      CALL LINE(COORD(1,NCORD1),SSSS(1,2),NPT,1,1)                      RADL1170
      CALL LINE(COORD(1,NCORD1),SSSS(1,2),NPT,1,1)                      RADL1180
```

```
      CALL LINE(COORD(1,NCORD1),SSSS(1,3),NPT,1,1)                      RADL1190
      CALL LINE(COORD(1,NCORD1),SSSS(1,3),NPT,1,1)                      RADL1200
      CALL LINE(COORD(1,NCORD1),TEMP,NPT,1,1)                           RADL1210
      CALL LINE(COORD(1,NCORD1),TEMP,NPT,1,1)                           RADL1220
      DO 200 I=1,3                                                      RADL1230
      BCDI(1)=BCD(I)                                                    RADL1240
      DO 200 J=1,NPT                                                    RADL1250
      XN=COORD(J,NCORD1)-SP(I)                                          RADL1260
      YN=SSSS(J,I)-SP(I)                                                RADL1270
      CALL SYMBOL(XN,YN,0.14,BCDI,0.0,1)                                RADL1280
  200 CALL SYMBOL(XN,YN,0.14,BCDI,0.0,1)                                RADL1290
      BCDI(1)=BCD(4)                                                    RADL1300
      DO 250 I=1,NPT                                                    RADL1310
      XN=COORD(I,NCORD1)-SP(4)                                          RADL1320
      YN=TEMP(I)-SP(4)                                                  RADL1330
      CALL SYMBOL(XN,YN,0.14,BCDI,0.0,1)                                RADL1340
  250 CALL SYMBOL(XN,YN,0.14,BCDI,0.0,1)                                RADL1350
      CALL SYMBOL(0.5,-0.75,0.14,'SIGMA R - DIAMOND  SIGMA T - ASTERISK'RADL1360
     1,0.0,37)                                                          RADL1370
      CALL SYMBOL(0.5,-0.75,0.14,'SIGMA R - DIAMOND  SIGMA T - ASTERISK'RADL1380
     1,0.0,37)                                                          RADL1390
      CALL SYMBOL(0.5,-1.0,0.14,'SIGMA Z - CROSS  TEMPERATURE - SQUARE',RADL1400
     10.0,37)                                                           RADL1410
      CALL SYMBOL(0.5,-1.0,0.14,'SIGMA Z - CROSS  TEMPERATURE - SQUARE',RADL1420
     10.0,37)                                                           RADL1430
      CALL SYMBOL(0.5,-1.25,0.07,'STRESS SCALE =      X 10   PSI/INCH',0RADL1440
     1.0,35)                                                            RADL1450
      CALL SYMBOL(0.5,-1.25,0.07,'STRESS SCALE =      X 10   PSI/INCH',0RADL1460
     1.0,35)                                                            RADL1470
      CALL ANN(DS,DSS,KEX)                                              RADL1480
      PEX=KEX                                                           RADL1490
      CALL NUMBER(1.375,-1.25,0.07,DSS,0.0,3)                           RADL1500
      CALL NUMBER(1.375,-1.25,0.07,DSS,0.0,3)                           RADL1510
      CALL NUMBER(1.880,-1.18,0.07,PEX,0.0,-1)                          RADL1520
      CALL NUMBER(1.880,-1.18,0.07,PEX,0.0,-1)                          RADL1530
      CALL SYMBOL(0.5,-1.5,0.07,'TEMPERATURE SCALE =       DEGREES/INCH RADL1540
     1',0.0,38)                                                         RADL1550
      CALL SYMBOL(0.5,-1.5,0.07,'TEMPERATURE SCALE =       DEGREES/INCH RADL1560
     1',0.0,38)                                                         RADL1570
      CALL NUMBER(1.6875,-1.5,0.07,DT,0.0,1)                            RADL1580
      CALL NUMBER(1.6875,-1.5,0.07,DT,0.0,1)                            RADL1590
      CALL SYMBOL(0.5,-1.75,0.07,'RADIAL POSITION SCALE =      INCHES/INRADL1600
     1CH',0.0,40)                                                       RADL1610
      CALL SYMBOL(0.5,-1.75,0.07,'RADIAL POSITION SCALE =      INCHES/INRADL1620
     1CH',0.0,40)                                                       RADL1630
      CALL NUMBER(1.9375,-1.75,0.07,DX,0.0,2)                           RADL1640
      CALL NUMBER(1.9375,-1.75,0.07,DX,0.0,2)                           RADL1650
      CALL SYMBOL(1.125,6.75,0.14,'STRESS AND TEMPERATURE',0.0,22)      RADL1660
```

```
      CALL SYMBOL(1.125,6.75,0.14,'STRESS AND TEMPERATURE',0.0,22)      RADL1670
      CALL SYMBOL(1.125,6.5,0.14,'VERSUS RADIAL POSITION',0.0,22)       RADL1680
      CALL SYMBOL(1.125,6.5,0.14,'VERSUS RADIAL POSITION',0.0,22)       RADL1690
      CALL SYMBOL(1.125,6.25,0.14,'TIME = ',0.0,7)                      RADL1700
      CALL SYMBOL(1.125,6.25,0.14,'TIME = ',0.0,7)                      RADL1710
      CALL NUMBER(2.00,6.25,0.14,TIME,0.0,3)                            RADL1720
      CALL NUMBER(2.00,6.25,0.14,TIME,0.0,3)                            RADL1730
      CALL SYMBOL(2.625,6.25,0.14,TITLE,0.0,12)                         RADL1740
      CALL SYMBOL(2.625,6.25,0.14,TITLE,0.0,12)                         RADL1750
      CALL PLOT(0.0,13.0,-3)                                            RADL1760
      CALL PLOTE                                                        RADL1770
      WRITE(6,1600)                                                     RADL1780
      RETURN                                                            RADL1790
 1000 FORMAT('1',38X,'TEMPERATURE AND STRESS VALUES, TIME = ',F10.5,//,1RADL1800
     13X,'NODE',5X,'COORDINATE',8X,'TEMPERATURE',10X,'SIGMA',14X,'TAU', RADL1810
     2/,25X,'R/T/Z',31X,'R/T/Z',12X,'RT/TZ/ZR',/)                       RADL1820
 1001 FORMAT(' ',10X,I5,2P5G18.6)                                       RADL1830
 1002 FORMAT('0',15X,1P1G18.6,18X,1P3G18.6)                             RADL1840
 1003 FORMAT(' ',15X,1P1G18.6,18X,1P3G18.6)                             RADL1850
 1600 FORMAT('0',25X,'*** GRAPHS HAVE BEEN DRAWN ***')                  RADL1860
 2000 FORMAT('1',53X,'STRESS VERSUS PADIUS',//,44X,'SIGMA R - "."','SIGMRADL1870
     1A T - "+"','SIGMA Z - "*"',///)                                   RADL1880
 2001 FORMAT('1',51X,'TEMPERATURE VERSUS RADIUS',///)                   RADL1890
      END                                                               RADL1900
```

LIST OF REFERENCES

1. McClanahan, E.D., Busch, R., Fairbanks, J., and Patten, J.W., Initial Work on the Application of Protective Coatings to Marine Gas Turbine Components by High-Rate Sputtering, ASME Paper No. 74-GT-100, 1974.

2. Parker, E.R., Materials Data Book For Engineers and Scientists, McGraw-Hill, 1967.

3. Materials Selector 73, Materials Engineering, Vol. 76, No. 4, Reinhold Publications, September, 1972.

4. Holman, J.P., Heat Transfer, 3rd ed., McGraw-Hill, 1972.

5. Kreith, F., Principles of Heat Transfer, pg. 466, 3rd ed., Intext Educational Publishers, 1973.

6. Krall, K.M. and Eckert, E.R.G., "Local Heat Transfer Around a Cylinder at Low Reynolds Number", Journal of Heat Transfer, pg. 273-274, May, 1973.

7. Lew, G.T., A Three Dimensional Solution of the Transient Field Problem Using Isoparametric Finite Elements, MSME Thesis, Naval Postgraduate School, Monterey, California, 1972.

8. Leonidas, E., A General Purpose Three Dimensional Stress Analysis Program, MSME Thesis, Naval Postgraduate School, Monterey, California, 1971.

9. Zienkiewicz, O.C., The Finite Element Method in Engineering Science, McGraw-Hill, 1971.

10. Hanson, D.E., Three-Dimensional Elasto-Static Problems Using Isoparametric Finite Elements, MSME Thesis, Naval Postgraduate School, Monterey, California, 1971.

11. Naval Postgraduate School Report 59C172121A, Three Dimensional Finite Element Studies Part One: Service Routines, by Cantin, G., December, 1972.

12. Users Manual, W.R. Church Computer Center, Naval Postgraduate School, Monterey, California, 1975.

INITIAL DISTRIBUTION LIST

		No. Copies
1.	Defense Documentation Center Cameron Station Alexandria, Virginia 22314	2
2.	Library, Code 0212 Naval Postgraduate School Monterey, California 93940	2
3.	Department Chairman, Code 59Fu Department of Mechanical Engineering Naval Postgraduate School Monterey, California 93940	2
4.	Professor Paul F. Pucci, Code 59Pc Department of Mechanical Engineering Naval Postgraduate School Monterey, California 93940	5
5.	Professor Gilles Cantin, Code 59Ci Department of Mechanical Engineering Naval Postgraduate School Monterey, California 93940	5
6.	Professor Craig Comstock, Code 53Zk Department of Mathematics Naval Postgraduate School Monterey, California 93940	1
7.	Assoc. Professor David Salinas, Code 59Zc Department of Mechanical Engineering Naval Postgraduate School Monterey, California 93940	1
8.	Lt. Richard B. Bubeck Production Department Mare Island Naval Shipyard Vallejo, California 94592	2

Thesis
B82943 Bubeck 162282
c.1
Transient thermal stress analysis of composite structures including continuously varying properties.